Getting Started with Metamask!

Markus Viola Follow
Feb 12 · 12 min read

In this tutorial, we'll be discussing about an essential technology that correlates with the core operation of Ethereum: **transactions**. Whenever you have to make transactions, you will and will always have to go through that fox dude above.

What's the good news...

It's a great that you're here, because I'm gonna discuss about the basic stuffs on how Metamask works, just enough for you to get started on your dApp developments.

Understanding Metamask

> Metamask is a bridge that allows you to visit the distributed web of tomorrow in your browser today. It allows you to run Ethereum DApps right in your browser without running a full Ethereum node.

Basically, when you do transactions on the blockchain network, these uses computing power which what we call, *gas cost*. And much like gas in real life, you have to pay for it. So in order for you to pay that *gas cost*, you will need a wallet that handles your money, and that's where Metamask comes in.

Well, I guess that's enough information about Metamask for now, we don't have to dive in to every side of the box, I bet you get the gist of its purpose already. So without further ado, let's jump right into the first question!

How can I install it?

Before you could actually install Metamask, you have to know that Metamask only supports a couple of browsers. Metamask supports the following browsers:

- **Google Chrome** (or any Chromium browsers, *Brave*, a DApp Chromium browser.)
- **Mozilla Firefox**
- **Opera**

If you already have any of the above, you can proceed, otherwise, you should download at least one that is listed browsers and you're good to go!

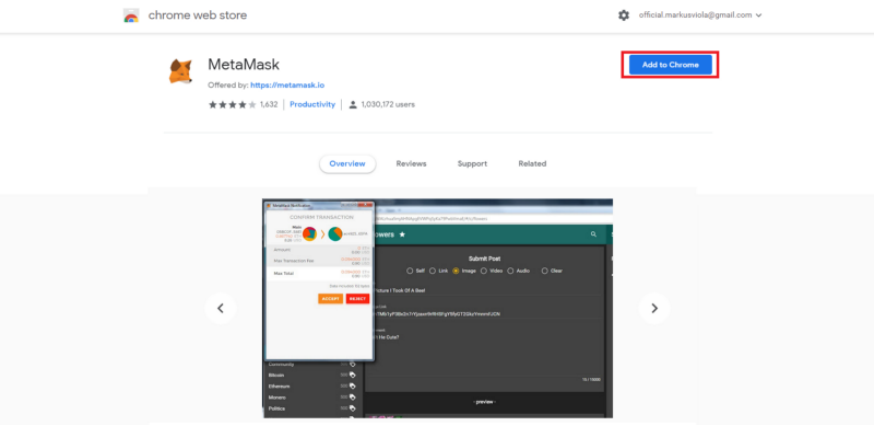
So, for you to download Metamask, you should first go to this website:

Metamask: https://metamask.io/

You can also download Metamask on your browser's plugin extension web store, but if you want to be directed automatically to the download page dedicated for your browser, this is the way to go.

Once you're inside, you should see Metamask's iconic fox head following your mouse direction. Under it, you should see the **GET CHROME EXTENSION** orange button, click that, and you'll be automatically directed to your browser's store.

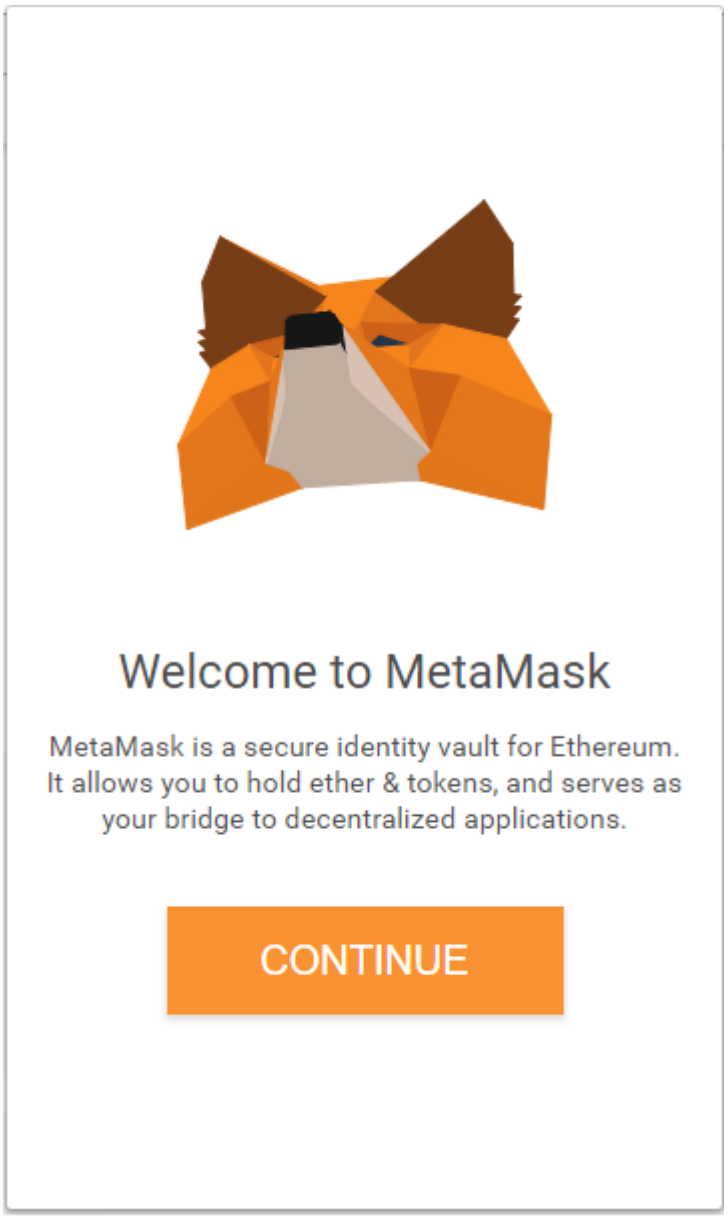
For Google Chrome browser's example, you should see something like this:



Now after getting in, just add the plugin extension, and it will install itself on the browser after downloading. (in Chrome's case, click the *Add to Chrome* button.)

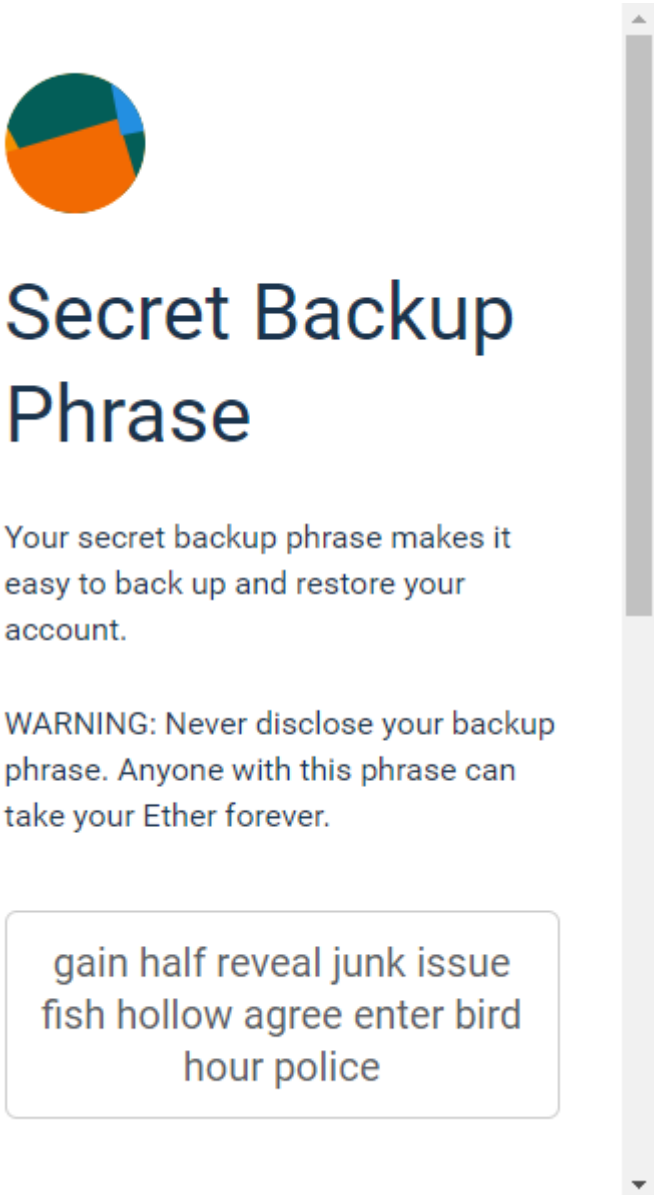
Once it is installed, a new tab will suddenly pop up, it's the Metamask wallet's welcome page. But you'll get the same interface on the actual wallet so just close it. Once closed, go and click here:

If you click that icon, you should trigger the Metamask wallet's user interface. This is how the wallet looks like:



Now just click continue, and you'll be directed to creating your own password. Don't go **import seed** since you don't have an existing account yet. You will use that if you want to recover your account which will be tackled as well later on.

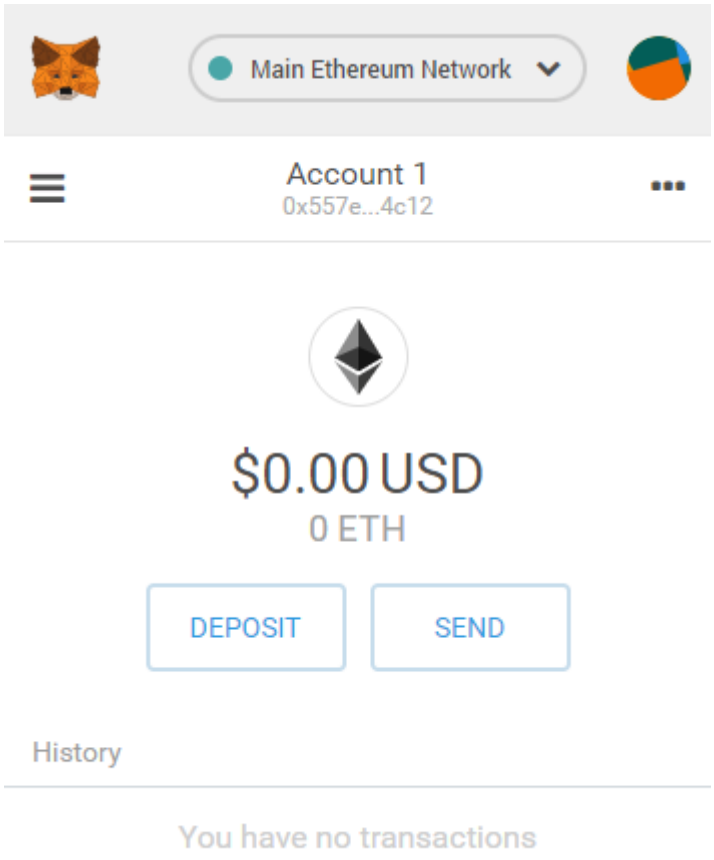
After creating and confirming your password, you'll go through some terms and agreements, then you'll be given a **secret backup phrase**. That phrase is the heart of your account since it is derived from a unique hash created when you made created the account just now.



As said, you shouldn't compromise the phrase to anyone, because if you're wallet has cryptocurrencies inside then that can be transferred to someone else. Although Metamask is a wallet made for developers, you still have to exercise secrecy.

(The account I have shown is for demonstration purposes so it's totally fine.)

After saving the phase, you will need to confirm it next. Then it will ask you if you want to deposit, skip, and close that for now.



Finally, you're done! You have successfully installed Metamask, and made you're very own wallet. Yeah, we did install Metamask, but what is the catch about *that* wallet, what is it capable of. This leads us to the next point:

What can it do?

For the features of Metamask, there's actually a lot that you can do, but I'm not gonna cover everything since my focus is to teach you just enough to get you ready for developing your Ethereum decentralized applications.

Here are the basic features of Metamask that you will need:

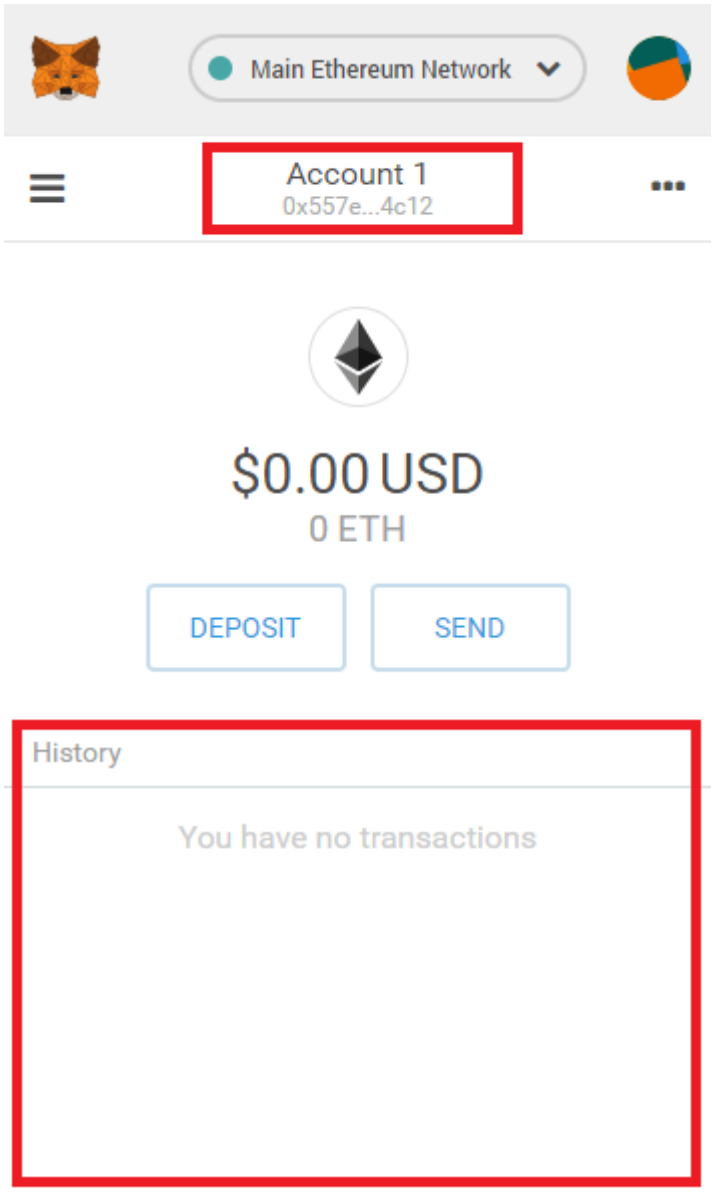
- **Add/Importing accounts**
- **Switching accounts**
- **Switch networks**
- **Request tokens** (Testnet Faucet)

- **Sending tokens** (Cryptocurrency)
- **Add/Search token**
- **Get backup seed**

Alright, so that's all the things that you will learn about in this tutorial. Let's now head to the next question:

How do I use it?

To track our progress easier, let's just learn all those listed features from top to bottom. But before that, I'm just gonna show you a few helpful things:



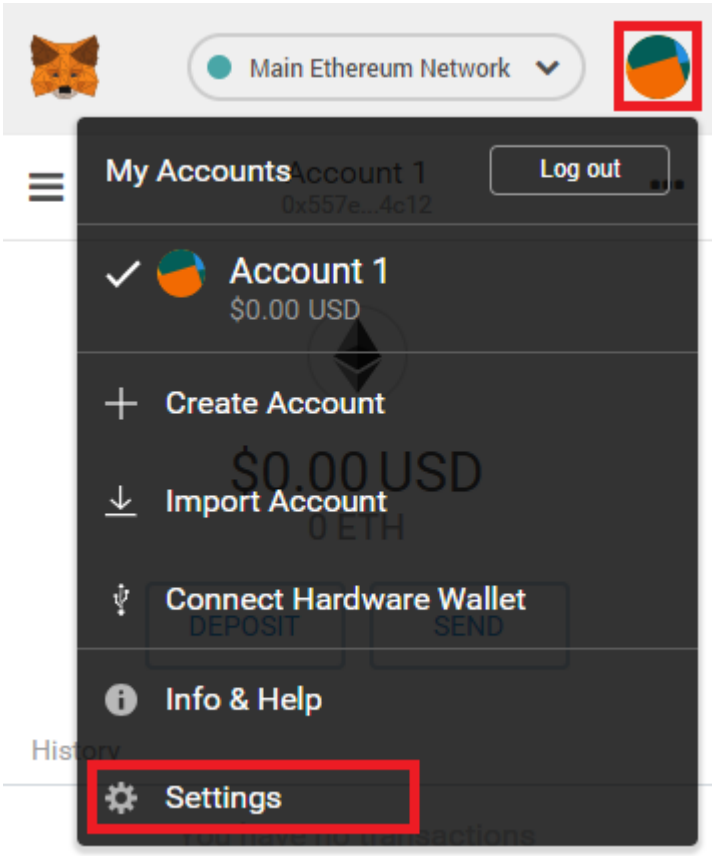
Let's try to look at the highlighted portion on the top side.

Below the name of the account: *Account 1*, you can see random-looking character string, something that looks like this, **0x557e...4c12**. That is your account's wallet address, if you click it, the address will be copied on your clipboard.

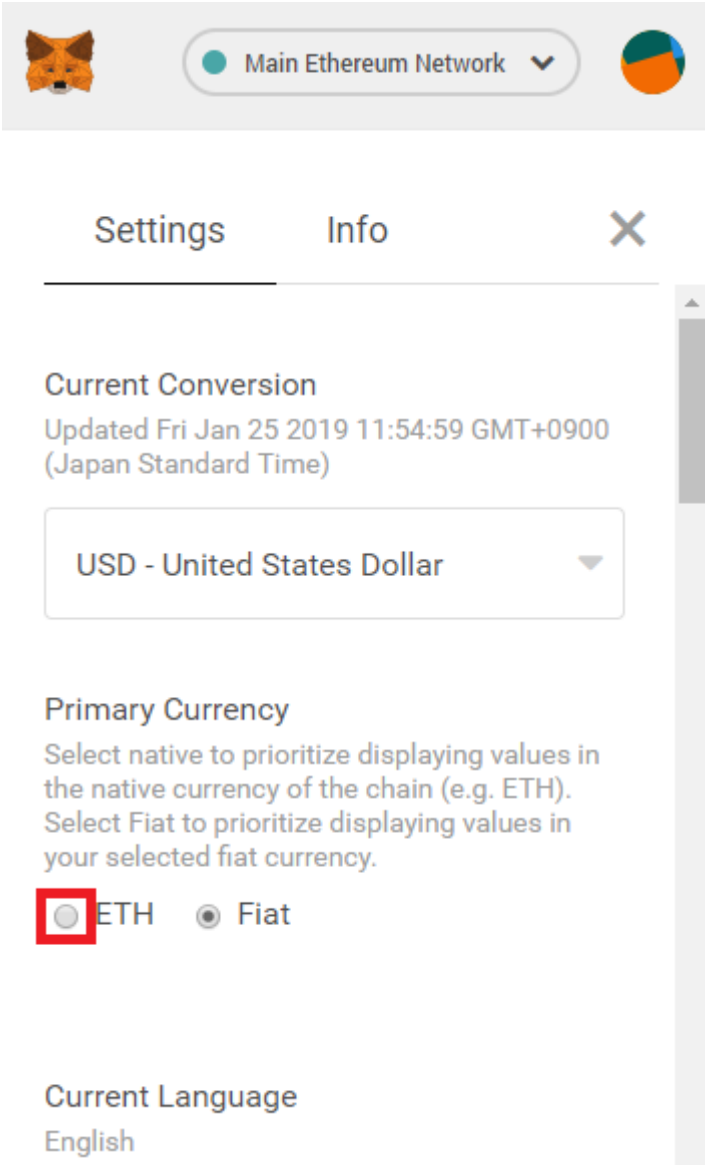
This is very helpful since you'll be using your account a lot on your smart contract development and on your dApps as a test account.

Next is the one below, that is your transaction history. There, you will see the state of your transactions whether or not it is finished, pending or the transaction had failed, etc.

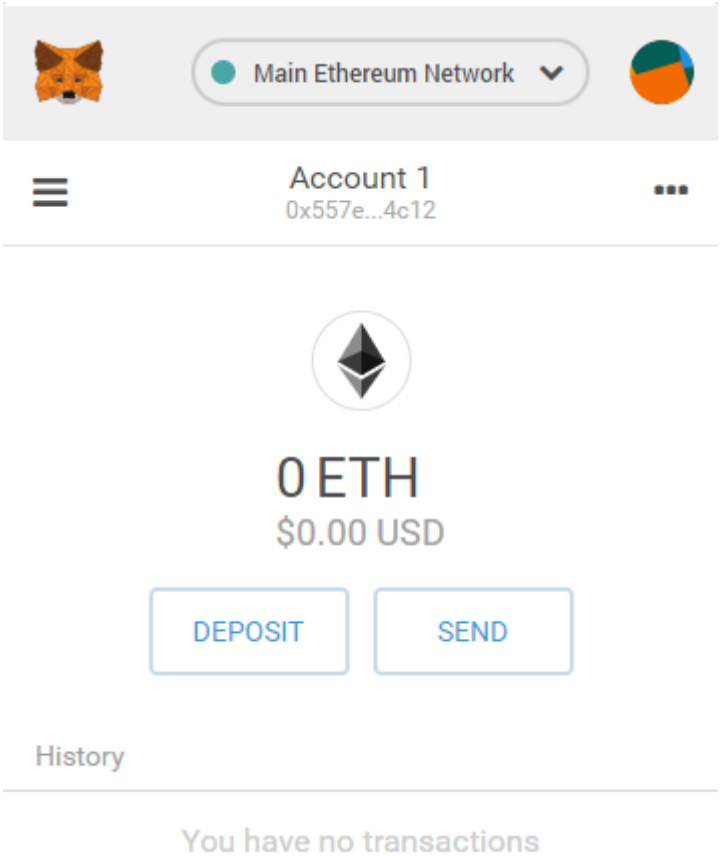
Also, if you want to switch the value from Fiat to ETH you can do that by going to the top right highlighted button, as shown here:



When clicked, a window will show up under it. From there, you can see from the bottom most part, which is the **Settings** button with the gear icon. Hit that button and it should this:



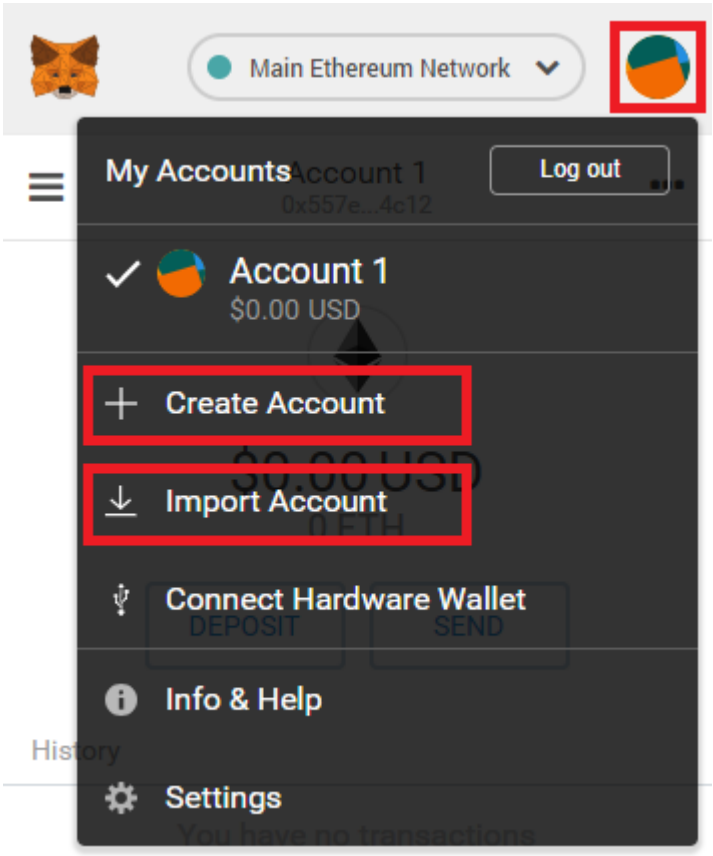
You'll be taken to this interface, initally on the *Settings* tab. Without scrolling, you'll be able to see the *Primary Currency* section, from the *Fiat* radio button, switch it to *ETH, and that's it. If you go back to your Main interface by hitting the fox head button, it should change by now:



So, too much for that, now let's head on to the first topic.

Add/Importing accounts

For this feature, this lets you add more than one wallet. This can be useful both in development and in real life. Since you'll be making applications, you will need to test not only one account, you have to get your accounts interact with each other on a smart contract, makes sense right?



Adding accounts

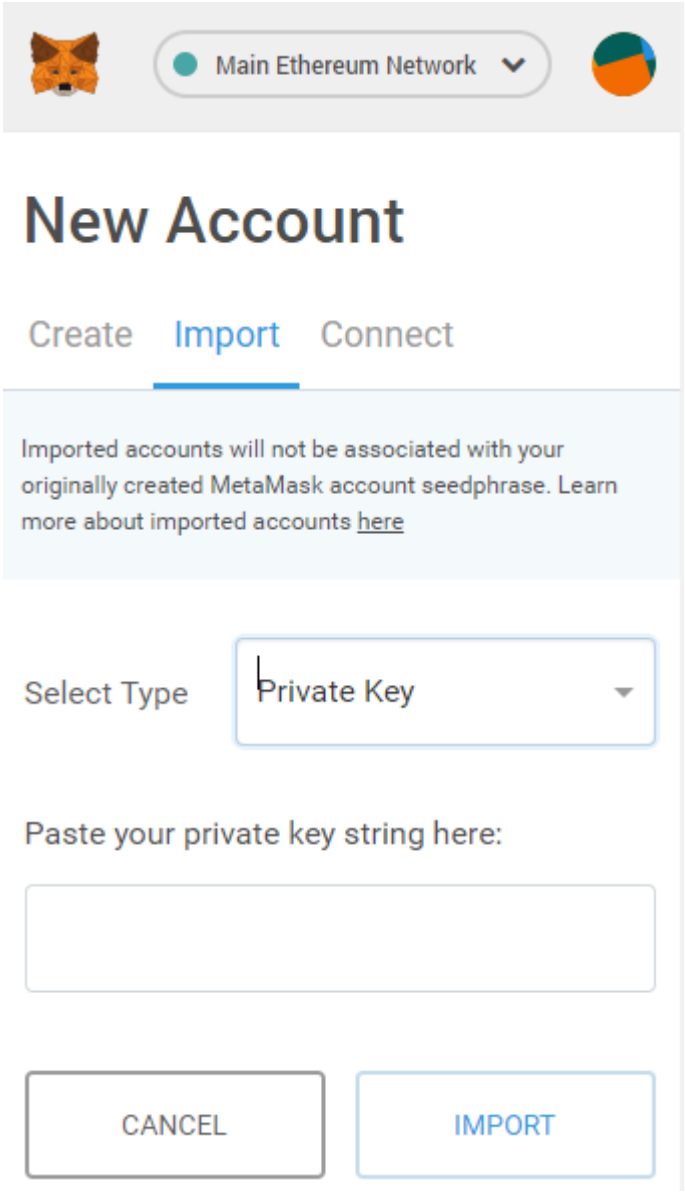
From main, for you to add an account, just go to the highlighted top right button and click it. This will prompt a new window, showing a list of items.

On the top side of that window, it shows the list of accounts that you currently have. As of now, you only have one account which is *Account 1*.

To add another, just click the first option below it: **+ Create Account**. When clicked, you'll be directed to a very self explanatory interface, which contains an input for the **name your account** and the button **Create**, hit it.

When you go back, alas, you have your new account! It also has its own unique address, which is nice. You can make as much as you want. (I think? Haven't tried 100 accounts yet.)

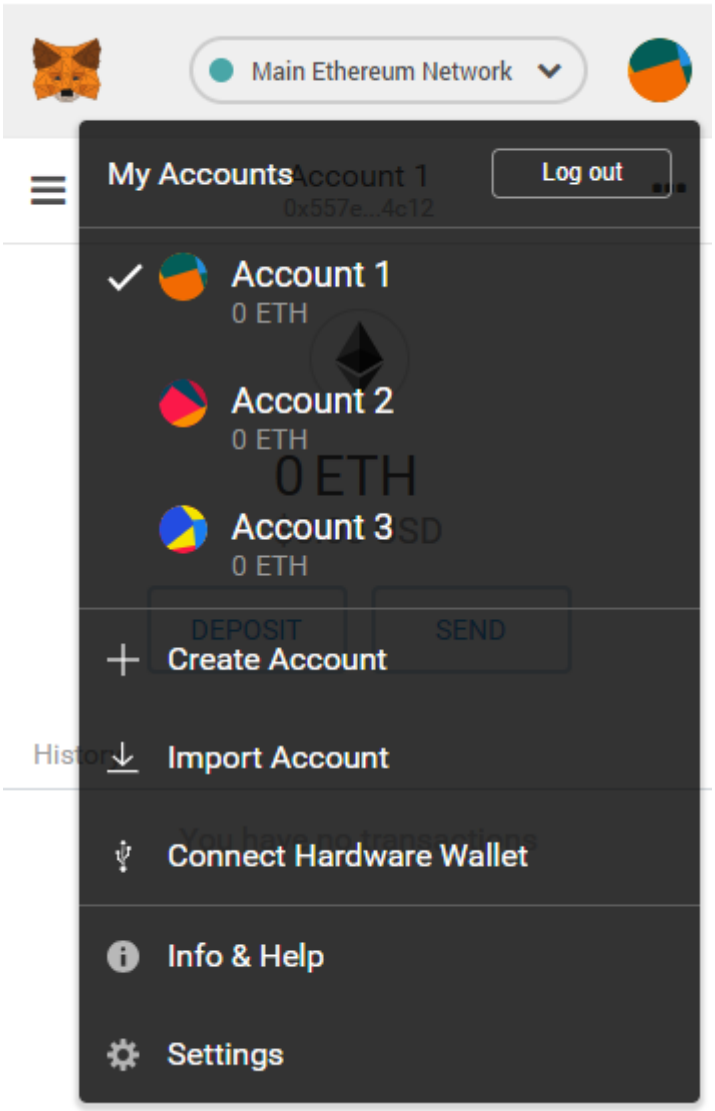
Importing accounts



If you have an existing account, you can hit the **Import Account**. You will be taken in an interface where you will have to put a *secret backup phase* in the private key string text input of the *Private Key* category. If you want to import JSON, you can do so by changing it in JSON. But it's very unlikely to use that one though. Last thing to do is to hit the **Import** button.

Switching accounts

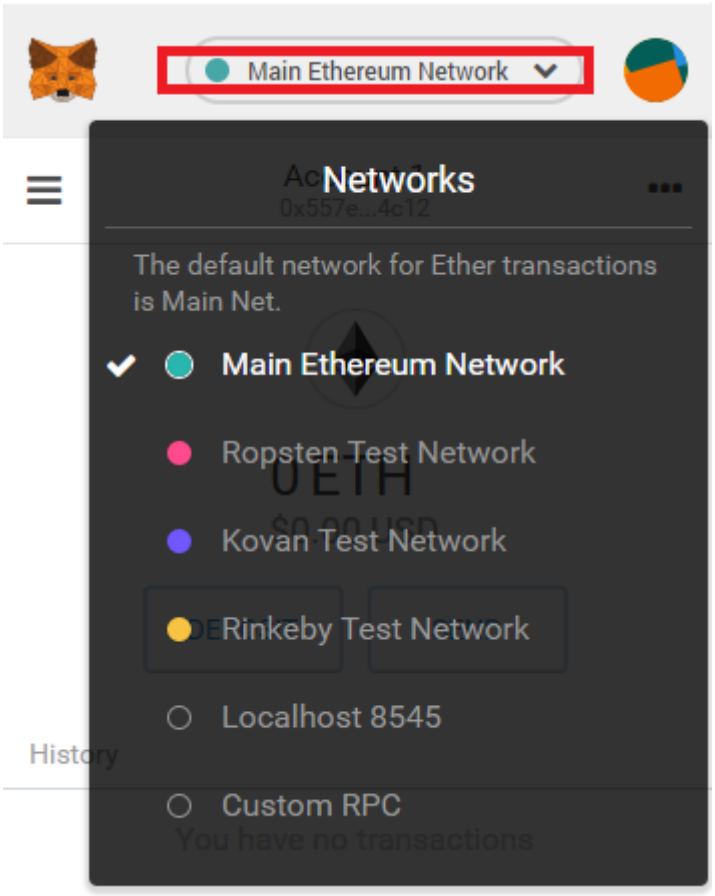
Since you learned how to import and add accounts, I assume you made new ones too. So let's learn how to switch accounts!



In the example above, you can see that I have already added 2 more accounts: *Account 2* and *Account 3*. On the same window as before, if we want to switch to *Account 2*, from the list, just click it. That easy, you have already switched accounts!

Switch networks

Switching networks is an important thing about Metamask. As you have recently seen, we're using the **Main Ethereum Network**. But that's not the development network, there's a network for one.

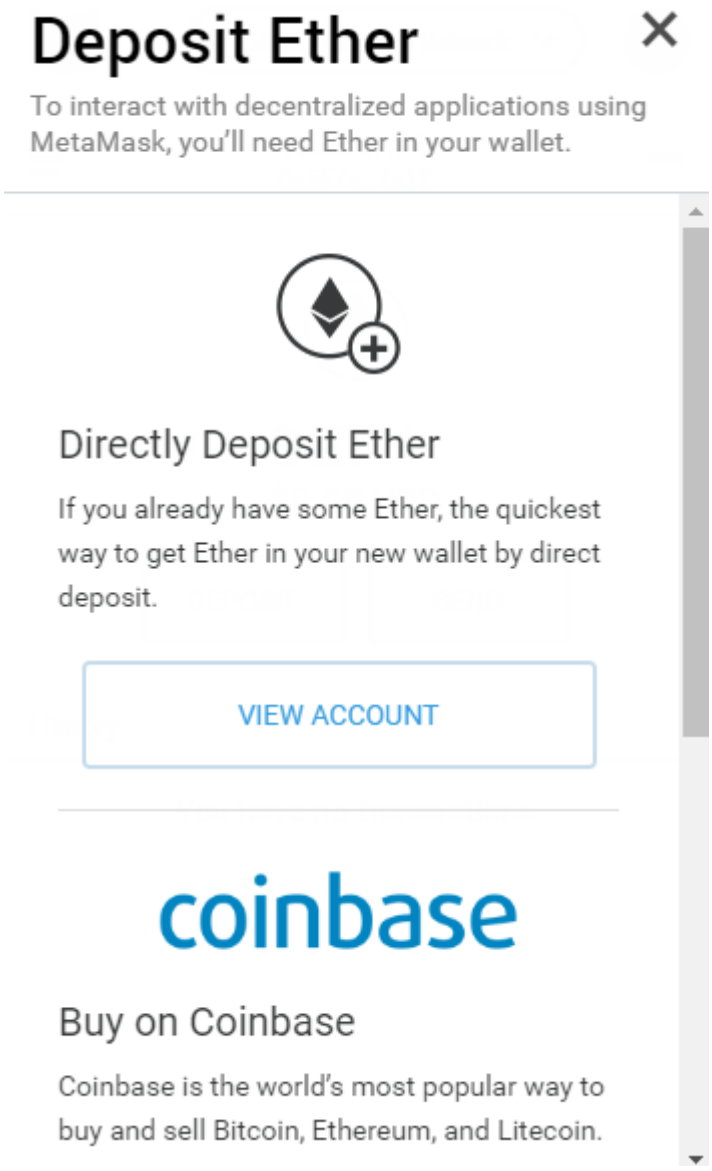


As you can see on the picture above, we're on the Main network. But underneath it are developers' available networks. Most often, developers usually use the **Ropsten Test Network**and **Rinkeby Test Network**. That is if you're using Remix directly with injected Web3 setting. *Localhost:8545* however, is for your local Ethereum network, if you're using Truffle and you made your own blockchain network locally.

Choosing a network is a case to case basis, but for a common action, let's choose **Ropsten Test Network**, click it and you're in it already!

Request tokens

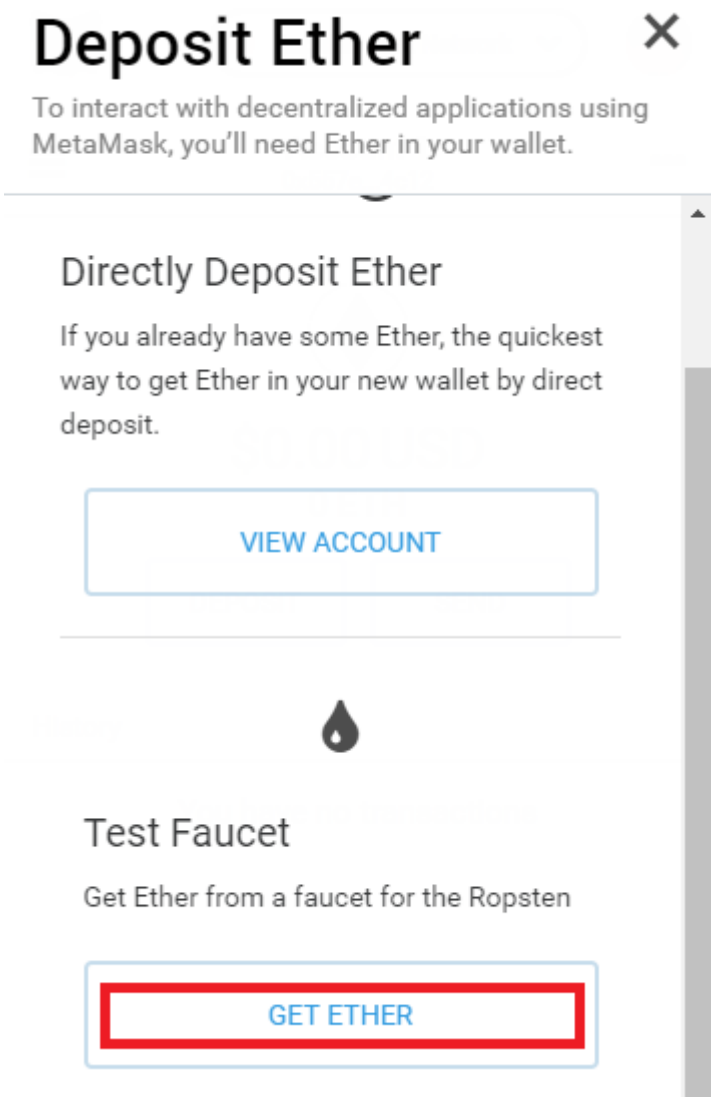
In making transactions, it costs gas amount, and gas costs money. So, we will need tokens everytime we interact/test contracts or sending tokens.



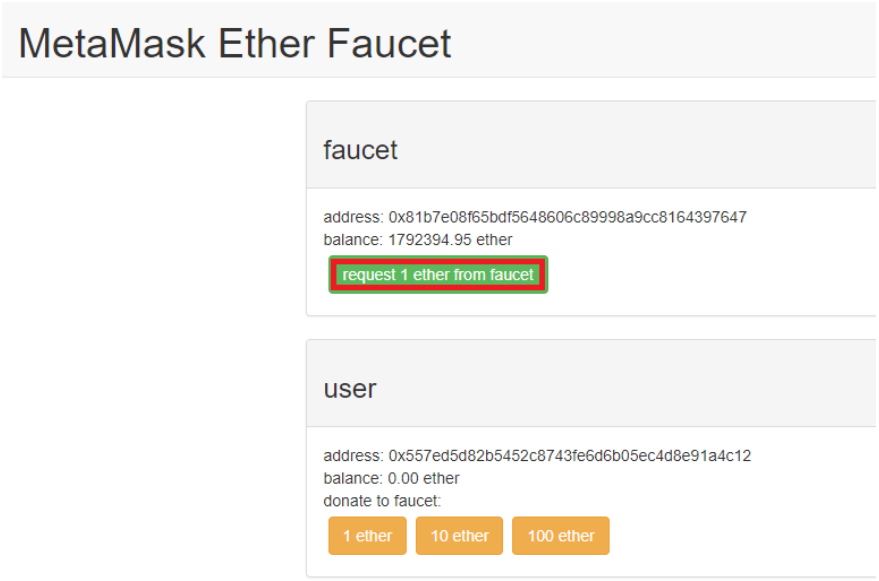
In the Main network, you have three ways to deposit token:

- **Directly Deposit Ether**: by clicking the *View Account*, using QR code, your other wallet that has tokens can transfer tokens by reading it with a camera.
- **CoinBase**: this is usually the way if you're new to cryptocurrency. You can go to its site, and buy ETH tokens.
- **ShapeShift**: basing on its name, it shifts your non-Ethereum tokens to Ethereum ones.

But all these are mostly used if you're actually going to spend and do real transactions on Ethereum. For the developers however, if you switch to a developer network such as **Ropsten Network**, you will see this:



Instead of Coinbase and ShapeShift, you will see a **Test Faucet**. By clicking the **Get Ether** button, you will be able to get free tokens. But these tokens have no real value since it's a testing network. You'll be taken on a website, for example for Ropsten.

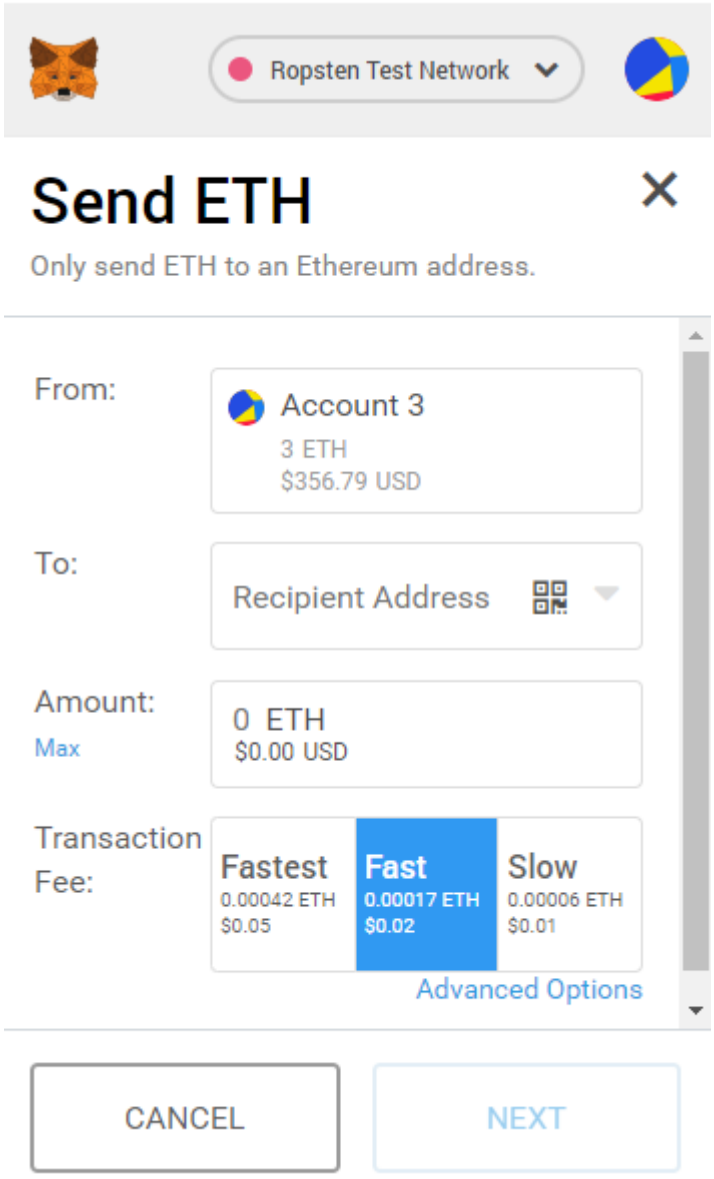


By just clicking the green button, you'll be able to request ether. From there, for a short moment, it will take effect and give you tokens that you'll be needing as gas for testing in development.

Here is the link of the Testnet Faucet, get yours now!

Send tokens

Before proceeding, try getting tokens for at your one account from the *Test Faucet* on *Ropsten Network*. If you already have tokens, we can now try send tokens to another account. To see the effect, send it to your other created account.

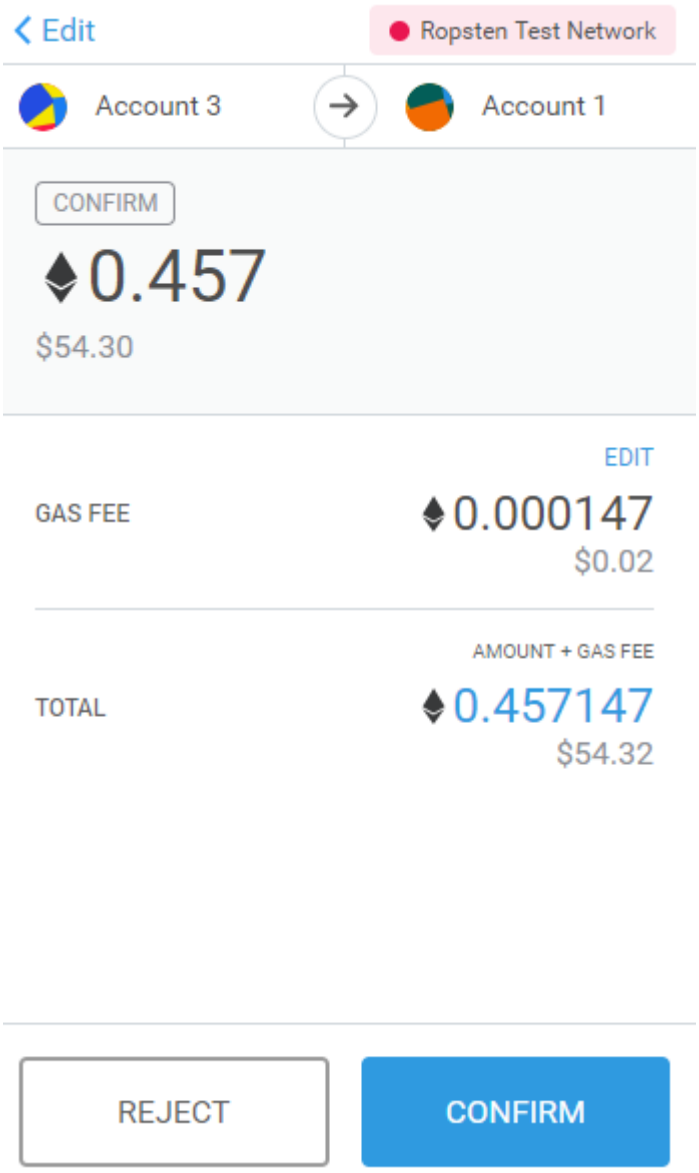


From the Ropsten Network, if you hit the **Send** button, you'll be sent to this interface. In the example, *Account 3* has 3 *ETH*, and by hitting the recipient address input text, you'll be able to choose or paste down an address.

In your case, use your account that has *ETH*, and send it to your other account. Set the amount on the **Amount** section, and you can set how fast the transaction will be beneath it, it's actually customizable if you choose *Advance Options*.

After you had set everything, hit **Next**. And you'll be sent in this interface:

Confirmation Interface



This is the exact interface whenever you make transactions on Ethereum using Metamask. You'll see a lot of this prompting interface once you start developing.

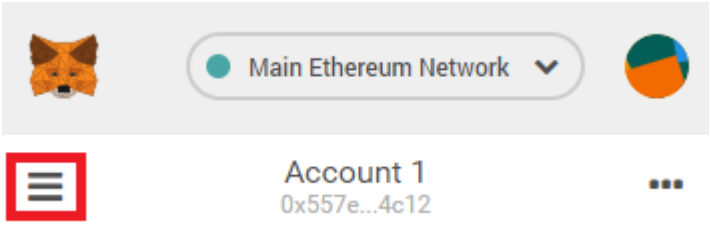
From here you can also edit the gas fee, to make your transaction faster. In here, we are sending 0.457 ETH to *Account 1*. Hit **Confirm** and you're transaction will be processed and will also show up on your *Transaction History* on the Main interface.

Add/Search tokens

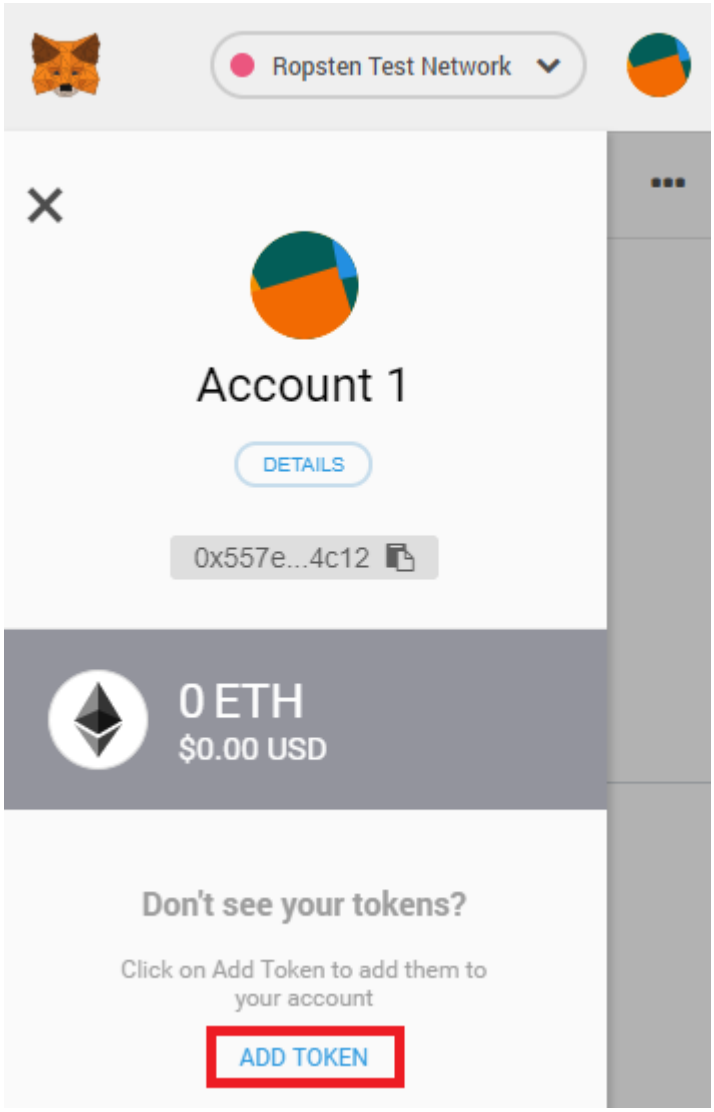
Aside from *ETH* token, there are also different kinds of tokens that you can use based on what kind of DApp are you using, usually when you make a DApp they also make their own cryptocurrency. So in here, I'll show you how to search and add your own token.

(By standard, cryptocurrency tokens should be ERC20 type.)

Add tokens

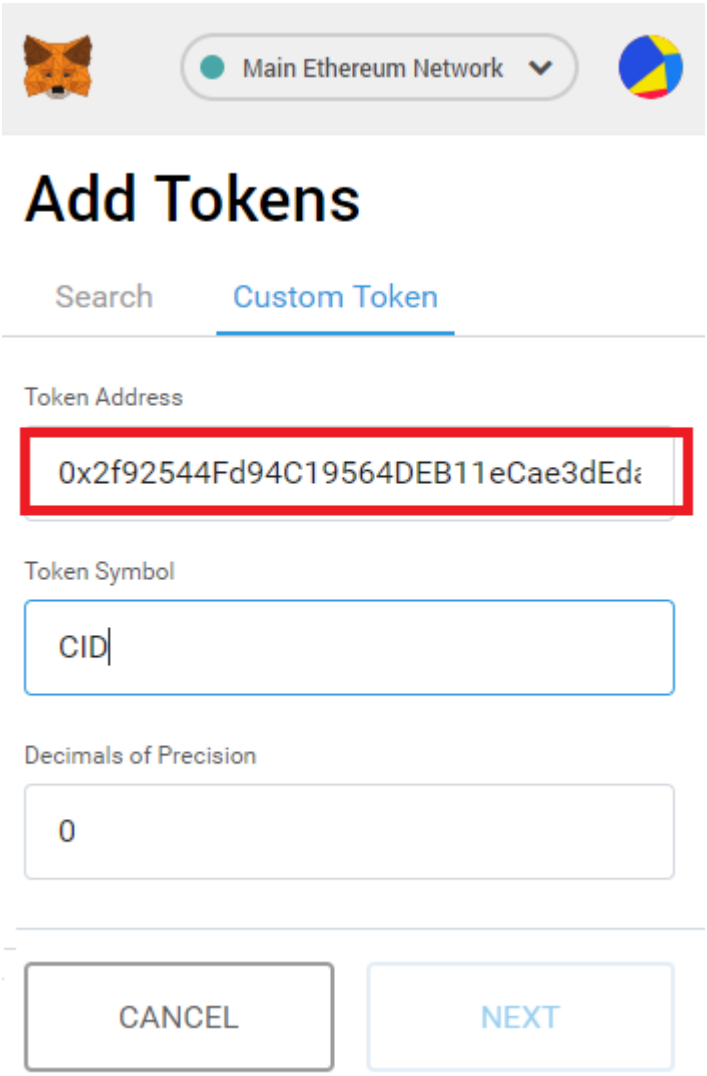


From the Main interface, go to top left menu icon beneath the fox's head or the Home button. Once you hit the menu, you'll be able to see this one slide:



From what you can see above, your token shows up, under that is actually a supposed list for the other tokens. Since you only have the *ETH* token, you can't notice the list.

With that said, you can add a token, it's either you add your own made token from an *ERC20* standard contract or you search someone's token assuming that you have one already. Let's go with the adding your own first.



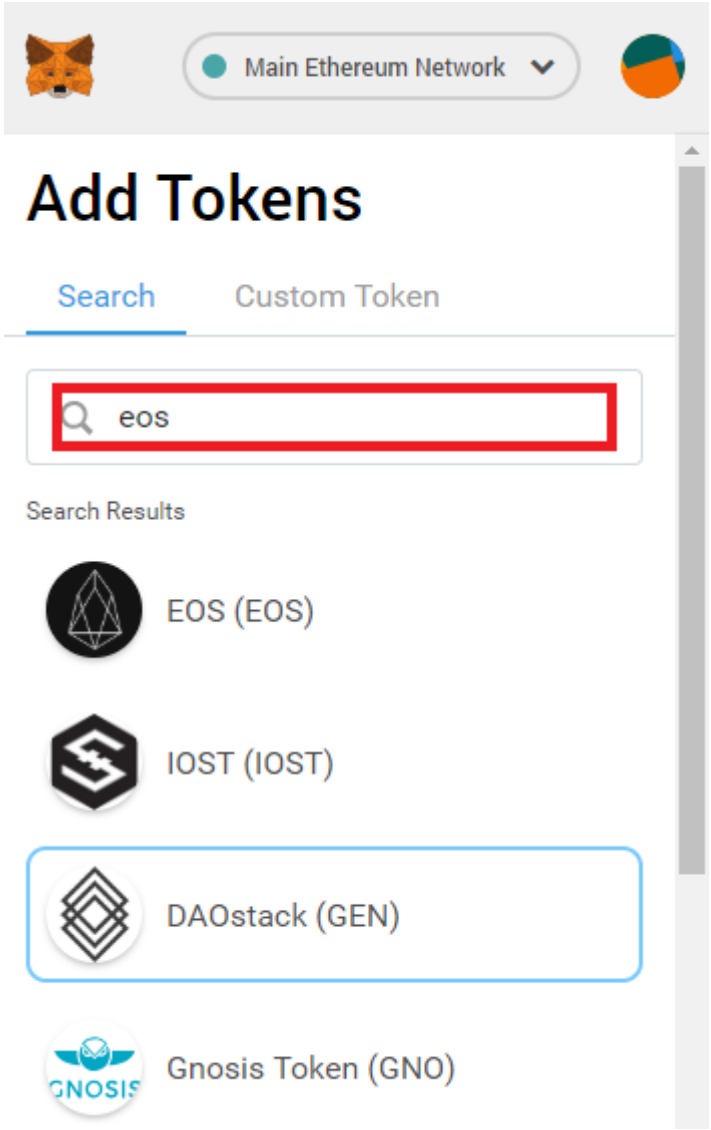
On the *Custom Token* tab, is where you can add your own created token. There are 3 input fields. The last two fills in by itself once you fill in the **Token Address** input text field. So the first one is the only needed.

If you plan to make your own token, you can try it on the Custom Token tab. Just hit **Next** then you're good to go, it is added to the list. If you choose one from the list, you can use it for your transactions related to that token.

Now on to the searching of token.

Search tokens

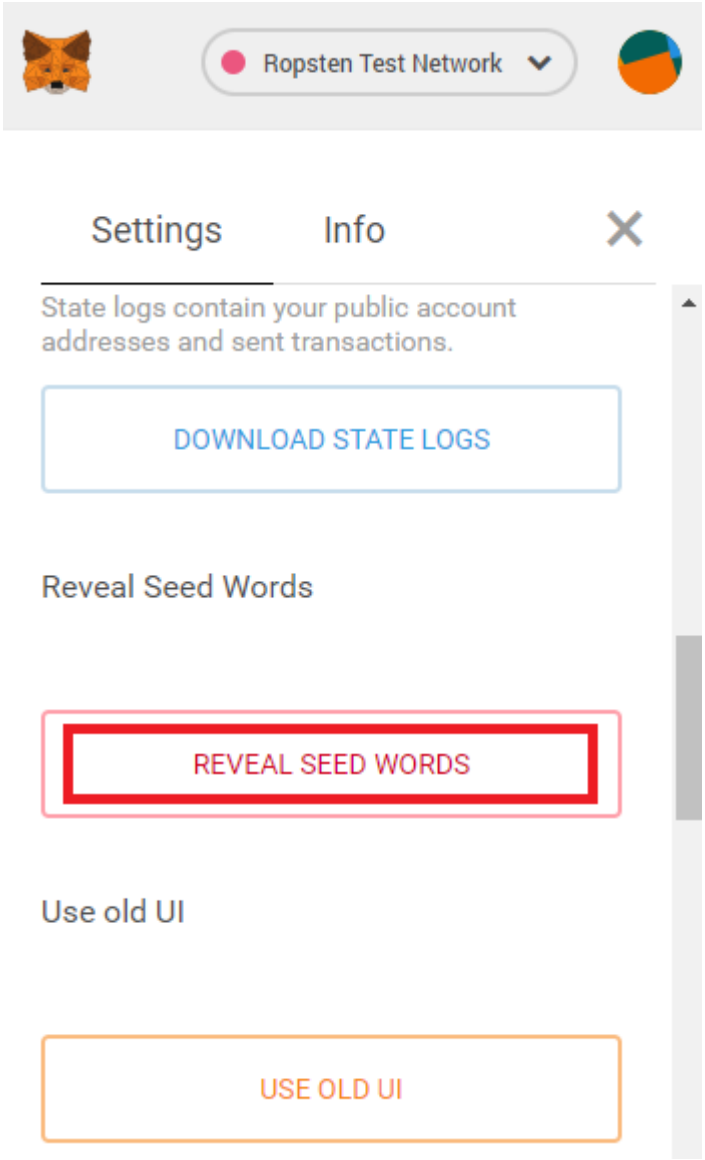
Same as before, if you go to the menu icon again, on the *Search* tab, you will see this interface:



As seen above, *EOS* is a known example of a crypto token. If in an instance, you have received *EOS* tokens, then you can search and verify it from here, and add it to your token list.

Get backup seed

If in case you forgot your *secret backup phrase*, you can get it from your settings, so go back to settings from the top right menu button.



When you get inside *Settings*, and if you scroll down on the middle you will find the *Reveal Seed Words* section. If you hit the **Reveal Seed Words** button, the *secret backup phrase* will be shown and you can copy it. Keep it safe!

Summary

It's finished! That tutorial sure was a bit long but I'm pretty sure that what you learned just now, will equip you enough for your future or

current developments for making your decentralized Ethereum applications.

In case you forget some things you learned from here, you can always refer back here. It's natural if you don't remember everything in one go, every developer's been there. You'll get better soon enough!

Look, even these non-blockchain related Microsoft developers are even happy for you:

But really now, you did a great job out there. Hope you learned a lot and good luck venturing more about blockchain.

Until then, signing out, and stay tuned!

---

openberry is a tutorial marketplace, designed to allow anyone to learn blockchain programming.

openberry | blockchain tutorial

Anyone using openberry can become a blockchain engineer. Openberry is a blockchain tutorial...

www.openberry.ac

openberry (@openberry_ac) | Twitter

The latest Tweets from openberry

(@openberry_ac): "https://t.co/o1leXC5rUL"

twitter.com