@metamask/snap-box¶

This repository is a starter template that combines theTypeScript Template Snap with aTruffle Box , for developing and testing snaps that interact with smart contracts.

MetaMask Snaps is a system that allows anyone to safely expand the capabilities of MetaMask. Asnap is a program that we run in an isolated environment that can customize the wallet experience.

Truffle Boxes are helpful boilerplates that allow you to focus on what makes your dapp unique. In addition to Truffle, Truffle Boxes can contain other helpful modules, Solidity contracts & libraries, front-end views and more; all the way up to complete example dapps. This box makes it easy to deploy and test contracts with a local Ganache instance, which can be used to test transaction insight snaps or smart contract account snaps.

Pre-requisites¶

Snaps should work with the latest LTS version of Node.js, but we recommend using the version specified in the.nvmrc file. If you usenvm you can easily switch to the right version by callingnvm use at the root of the project.

This box uses Yarn v3.x. If you are using Node 16 or later you can enable it withcorepack enable .

To interact with (your) snaps, you will need to installMetaMask Flask , a canary distribution for developers that provides access to upcoming features. You should install Flask in a separate browser profile from any existing MetaMask installation.

Before using this box, install Truffle and Ganache globally:

npm install -g truffle ganache

Installation¶

You can install this box with Truffle:

truffle unbox metamask/snap-box Alternatively, you can clone the snap-box repositoryusing this GitHub template .

Setup¶

You can add your own Infura API key to fork the Ethereum blockchain in your local instance of Ganache. Copy the.env.dist file inpackages/truffle/ to a new file.env and update theINFURA_PROJECT_ID variable with your API key.

Then, you can take your Secret Recovery Phrase from MetaMask Flask and put it as theMNEMONIC_PHRASE in this same file. This will make your first Ethereum account as the deployer of the test contracts in Ganache and provide you with a balance for the first 10 accounts for local testing.Remember, you should never share your Secret Recovery Phrase with anyone, and you should never use a Secret Recovery Phrase from any version of MetaMask that you use to custody real funds. This.env file is never and should never be uploaded to GitHub (it is explicitly excluded in.gitignore). This is only for testing locally with Ganache.

Alternatively, you can take the Mnemonic Phrase generated by Truffle and use that as your Secret Recovery Phrase when setting up MetaMask Flask. This will guarantee that your transaction history is empty each time you start testing.

Then, setup the development environment from the main directory:

yarn install &&

yarn start You are now ready to start modifying the packages to build your snap. You can put smart contracts inpackages/truffle/contracts and modify the files inpackages/snap andpackages/site . Connect MetaMask to your local Ganache instance by going into your settings and changing the network to "Localhost 8545" (if you cannot see it, make sure to click "Show test networks").