Truffle Suite

Update : Since this tutorial was published, we have releasedGanache , a personal blockchain and a replacement to the TestRPC. We have left this tutorial unaltered, but we highly recommend checking out ourGanache Documentation This article was originally written by Dan Finlay and published as agithub gist . Dan has kindly allowed us to republish it here!

If you're interested in building web apps withthe Ethereum blockchain , you may have foundthe Truffle web framework to be a nice fit for your needs.

For many types of Dapps (Distributed Apps), Truffle does everything you could want: It compiles your blockchain contracts, injects them into your web app, and can even run a test suite against them!

This is all great for you, butwhat about your users? Truffle has some great defaults for users who are willing to run a localEthereum JSON RPC server on their computer, but what about your users who just want to sign on and get started?

With Metamask, all your users need to do isinstall our Chrome plugin , and they will have their own secure blockchain accounts right there in the convenience of their browsers.

Metamask is just a Developer Preview right now, and has not been released to the general public. We don't recommend putting serious funds in it, but instead encourage you to use it to help prepare your dapps for Ethereum browsers.

It's possible your Truffle Dapp is already compatible with Metamask, but if you're like me, you want to go through your project and see what it's like from your user's perspective.

Just in case you haven't used Truffle before, I'm going to start by describing how to scaffold a minimal Truffle dapp. If you already have one set up, feel free to scroll ahead!

Installing Truffle Dependencies¶

You're going to need to have installednode.js .

From there, you need to install truffle (npm install -g truffle).

Also, you're going to need to run a local blockchain RPC server to test and develop against. I recommend usingTestRPC , which you install by runningnpm install -g ethereumjs-testrpc .

Next let's make sure we have ourtestrpc running in the background. Open your terminal and run the commandtestrpc . That's all! It runs on port8545 by default, just like most Ethereum RPCs, and so does Truffle.

Quick Note on RPCs¶

Ethereum RPCs can run in a pre-authenticated mode, that allows you to perform transactions without providing any additional credentials. Obviously this would be a dangerous thing to do on the open web, but for local development, it's the default method thattestrpc provides.

When you first starttestrpc , You get a list of initial accounts. These accounts start funded with a whole lot of Ether, so you can practice exchanging funds between accounts.

When you open a normal Truffle Dapp, it takes advantage of your local authenticated rpc, and has access to its account list. The default Truffle dapp just uses the first account in the list.

Setting up a Simple Truffle Dapp¶

Next, let's generate a basic Truffle dapp. The default result oftruffle init is a simple example currency.

To get it up and running, run these commands:

mkdir my-money

Create a folder for your new dapp

cd

my-money

Move into that folder

truffle init

Initialize a default truffle project in that folder

truffle build

Compile the dapp

truffle migrate

Publish the dapp on the blockchain

truffle serve

Host your web interface on port 8080

We just deployed a simple alt-coin calledMetaCoin to our local blockchain, and it's available to our browser onhttp://127.0.0.1:8080 !

If you visit it, you'll see that by default this new Dapp template signs you in with the first account on yourtestrpc account list, which happens to be the same account that got pre-populated with 10k shiny new Metacoins! That's because when you rantruffle migrate , Truffle used your first account as the contract publisher, and the contract says to fund the creator's account with 10k coins.

If you don't see any Metacoin in this account, it may be because you have Metamask already installed. Jump down to "Switching between TestRPC and Metamask accounts" below.

You can now send those coins to any account you want, so let's now set up some Metamask accounts, and then we can fund them!

Setting up Metamask¶

Now you'll want to installMetamask from the Chrome store .

Metamask is currently not listed on the Chrome store, but you get that link because you're an early adopter who we want to support.

When first setting up Metamask, you'll need to provide a password (used to encrypt your wallet). It is then going to generate a unique twelve-word phrase that you'll eventually be able to use to recover all your accounts.We haven't done that yet because this is just a developer preview! Please don't use Metamask for important things yet!

By default you get three accounts, all named Wallie. Again, our developer preview is showing. Eventually we're planning to integraterich persona management via uPort !).

To use Metamask with your local development blockchain, you need to point it at yourtestrpc .

1. Open Metamask
2. Click the gear icon in the bottom left
3. Enter yourtestrpc
4. address into the RPC field. It's probablyhttp://127.0.0.1:8545/
5. .

Metamask will close at this point to restart itself.

You can now sign into the dapp with Metamask.

1. Open Metamask
2. Enter your password
3. Select an account
4. Reload the Dapp page (some dapps will notice when you change accounts, but this basic one doesn't)

You might notice you don't have any Metacoin in this account. That's a good thing! It means your new currency isn't going out to random accounts on the internet!

To send some Metacoin to one of your Metamask accounts, you're going to need that account's address.

To copy a Metamask account's address:

1. Open Metamask
2. Click the details arrow next to the account whose address you want.
3. Click the "Copy Address" button.

You now you have a copy of your address to send to!

Switching Between TestRPC and Metamask Accounts¶

When Metamask is installed, it intercepts calls and transactions to the Ethereum blockchain, and forwards them to its own RPC (which is currently on the test-network by default).

Since Metamask is managing ethereum connections when it's installed, our Chrome browser right now can't access ourtestrpc accounts, because those accounts are managed by thetestrpc node itself, so we need to open a browser without metamask.

The easiest way to do this is to open an incognito window. You can also open another browser.

Funding Your Metamask Account¶

Once you're connected to yourtestrpc accounts again, you can send your Metacoin to whoever you please, including your new Metamask accounts.

You might also notice that your new Metamask accounts don't have any ether in them. Thetestrpc initializes the blockchain with just its starting accounts having any balance, so to fund a Metamask account, you need to send some Ether to those accounts from one of yourtestrpc accounts. (In a future version of TestRPCthere will probably be better ways to do this).

Here's a simple shell command to tell yourtestrpc to send 1 ether to your Metamask account. Let's assume0x0f91747e3a5df28d81ab30b2d8216c93263c0cf3 is the first account in yourtestrpc list, and0xbbd220f66e989a493c4d48531ea1e283abc385de is one of your Metamask accounts.

1e18 is equal to 1 ether, which will be more than enough for our purposes.

curl -d '{"jsonrpc":"2.0","method":"eth_sendTransaction","params":[{"from":"0x0f91747e3a5df28d81ab30b2d8216c93263c0cf3", "to":"0xbbd220f66e989a493c4d48531ea1e283abc385de", "value": 1e18}], "id":1}'

-X POST http://127.0.0.1:8545/

Synchronizing Accounts Between Metamask And Testrpc¶

If you want to use testrpc/Metamask combination for development purposes, another route to get the accounts in Metamask to be useful with testrpc is to use the HD wallet generation that both support. If you start testrpc it creates 10 different addresses that can be recovered by the supplied mnemonic. The`testrpc output show the 10 initial accounts, the 10 private keys for the accounts and the mnemonic to recreate those accounts.

Available Accounts

(0) 0x0b488d56e55996ea868e6bc640a59d10c71ff4e7 ... (9) 0x4decace9430b32cfd72ed99656adf80ebf23931d

Private Keys

(0) cb99e7dfaf138af48dde4e863f6dca0ab0acfad29f95954f08796f962656d5c8 ... (9) 29f734be35466ed5a62d0c7387412bc73d533bc24197cdcf2b8bbac8a982b198

HD Wallet

Mnemonic: rubber negative firm purity helmet barely six asset imitate nephew october pluck Base HD Path: m/44'/60'/0'/0/{account_index} If you use that mnemonic in Metamask torestore an existing vault instead of creating a new one Metamask will use the same accounts. Testrpc also support a command line option to supply the mnemonic instead of using a random one. Starting it withtestrpc -m "rubber negative firm purity helmet barely six asset imitate nephew october pluck" will give you the same accounts each time. That way you don't have to setup Metamask every time to use a new mnemonic.

Sending Metacoin from Metamask: So Meta

Now if we connect to our Dapp via Metamask, we should see we have some MetaCoin, and if we look in our Metamask plugin, we should have 1 ether too!

It's important that we have ether, because it's not only a currency for trade, it's also the currency for processing transactions on the Ethereum blockchain. This processing fee is referred to as "gas".

Let's try sending some Metacoin from one of our Metamask accounts to another.

First select the account that has the Metacoin and Ether. Now click the details arrow for another account, and copy its address (COPY ADDR).

Paste the address into the Dapp window, along with how much Metacoin you'd like to send, and hitsend !

You should see a notification pop-up, notifying you that you have a transaction to approve in Metamask.

You can either click "Approve" on the notification, or open the Metamask pop-up and review the transaction there.

This is one way that Metamask is a safer Ethereum browsing experience than running your own RPC. While a pre-authenticated RPC automatically approves all requests sent to it, Metamask gives the user an opportunity to review and approve or reject each transaction that's requested by the dapp. Long term, making transactions easy to intelligently review is an important priority for Ethereum developers.

Wrapping Up¶

This has been a simple example, but hopefully shows how Truffle and Metamask work together. They are able to pretty much work out of the box! That's because Truffle helps you writeweb3 -compliant dapps. Web3 is a Javascript API declared by the core Ethereum team, and since it was standardized early, tools that take advantage of its common methods work well together.

This makes connecting to a Truffle Dapp with Metamask not much harder than connecting via your own RPC, but it'sa lot easier for your users.

That's because Metamask is a hyper-light client that doesn't replicate the entire blockchain locally, but itdoes let users manage their own accounts, so they can casually benefit from the security of private key management, while placing trust for block validation on Metamask's configured RPC provider.

We hope this has been a useful introduction to developing with Truffle and Metamask!

Please, leave a comment or question,start a discussion on github , orTweet at us on Twitter . We're trying to make Metamask the easiest tool to let anyone benefit from using Distributed Apps on the Ethereum Blockchain.