```
3.3-Testing.md
```

Testing

Now that we have designed our blockchain, let's try it out. First let's make sure it builds and installs:

```
$ sudo make install
```

Then, let's make sure it runs. We'll just run it in the foreground to see the output on standard out:

```
$ learncoind -debug -printtoconsole -regtest
```

If there are no errors, then we can stop it (CTRL+c), then run it in the background with:

```
$ learncoind -debug -daemon
```

Configuration File

Let's create a configuration file, we can start with this template from the Bitcoin wiki and we'll save it to ~/.learncoin/learncoin.conf:

Quick Primer on addnode vs connect ## Let's say for instance you use addnode=4.2.2.4 ## ## addnode will connect you to and tell you about the nodes connected to 4.2.2.4. In addition it will tell ## ## the other nodes connected to it that you exist so they can connect to you. ## connect will not do the above when you 'connect' to it. ## It will *only* connect you to 4.2.2.4 and no one else.## ## ## So if you're behind a firewall, or have other problems ## ## finding nodes, add some using 'addnode'. ## ## ##

1 of 4 2/5/19, 10:02 PM

```
## If you want to stay private, use 'connect' to only
                                                           ##
## connect to "trusted" nodes.
                                                           ##
##
                                                           ##
## If you run multiple nodes on a LAN, there's no need for ##
## all of them to open lots of connections. Instead
## 'connect' them all to one node that is port forwarded
                                                          ##
##
   and has lots of connections.
                                                           ##
        Thanks goes to [Noodle] on Freenode.
##
                                                           ##
# Use as many addnode= settings as you like to connect to specific peers
#addnode=69.164.218.197
#addnode=10.0.0.2:8333
# Alternatively use as many connect= settings as you like to connect ONLY to specific peers
#connect=69.164.218.197
#connect=10.0.0.1:8333
# Listening mode, enabled by default except when 'connect' is being used
#listen=1
# Maximum number of inbound+outbound connections.
#maxconnections=
# JSON-RPC options (for controlling a running Bitcoin/bitcoind process)
# server=1 tells Bitcoin-Qt and bitcoind to accept JSON-RPC commands
server=1
# Bind to given address to listen for JSON-RPC connections. Use [host]:port notation for IPv6.
# This option can be specified multiple times (default: bind to all interfaces)
#rpcbind=<addr>
# If no rpcpassword is set, rpc cookie auth is sought. The default `-rpccookiefile` name
# is .cookie and found in the `-datadir` being used for bitcoind. This option is typically used
# when the server and client are run as the same user.
# If not, you must set rpcuser and rpcpassword to secure the JSON-RPC API.
# The config option `rpcauth` can be added to server startup argument. It is set at initialization time
# using the output from the script in share/rpcauth/rpcauth.py after providing a username:
# ./share/rpcauth/rpcauth.py alice
# String to be appended to bitcoin.conf:
rpcauth=alice:f7efda5c189b999524f151318c0c86$d5b51b3beffbc02b724e5d095828e0bc8b2456e9ac8757ae3211a5d9b16a22
# Your password:
# DONT_USE_THIS_YOU_WILL_GET_ROBBED_8ak1gI25KFTvjovL3gAM967mies3E=
# On client-side, you add the normal user/password pair to send commands:
#rpcuser=alice
#rpcpassword=D0NT_USE_THIS_Y0U_WILL_GET_R0BBED_8ak1gI25KFTvjovL3gAM967mies3E=
# You can even add multiple entries of these to the server conf file, and client can use any of them:
rpcauth=bob:b2dd077cb54591a2f3139e69a897ac$4e71f08d48b4347cf8eff3815c0e25ae2e9a4340474079f55705f40574f4ec99
# How many seconds bitcoin will wait for a complete RPC HTTP request.
# after the HTTP connection is established.
#rpcclienttimeout=30
# By default, only RPC connections from localhost are allowed.
# Specify as many rpcallowip= settings as you like to allow connections from other hosts,
# either as a single IPv4/IPv6 or with a subnet specification.
# NOTE: opening up the RPC port to hosts outside your local trusted network is NOT RECOMMENDED,
```

2 of 4 2/5/19, 10:02 PM

```
# because the rpcpassword is transmitted over the network unencrypted.
# server=1 tells Bitcoin-Qt to accept JSON-RPC commands.
# it is also read by bitcoind to determine if RPC should be enabled
#rpcallowip=10.1.1.34/255.255.255.0
#rpcallowip=1.2.3.4/24
#rpcallowip=2001:db8:85a3:0:0:8a2e:370:7334/96
# Listen for RPC connections on this TCP port:
#rpcport=8332
# You can use Bitcoin or bitcoind to send commands to Bitcoin/bitcoind
# running on another host using this option:
#rpcconnect=127.0.0.1
# Wallet options
# Create transactions that have enough fees so they are likely to begin confirmation within n blocks
(default: 6).
# This setting is over-ridden by the -paytxfee option.
#txconfirmtarget=n
# Pay a transaction fee every time you send bitcoins.
#paytxfee=0.000x
# Miscellaneous options
# Pre-generate this many public/private key pairs, so wallet backups will be valid for
# both prior transactions and several dozen future transactions.
#keypool=100
# Enable pruning to reduce storage requirements by deleting old blocks.
# This mode is incompatible with -txindex and -rescan.
# 0 = default (no pruning).
# 1 = allows manual pruning via RPC.
\# >=550 = target to stay under in MiB.
#prune=550
# User interface options
# Start Bitcoin minimized
#min=1
# Minimize to the system tray
#minimizetotray=1
```

Note that we've enabled the RPC server with server=1.

Premine

Let's import the private key for our genesis block premine transaction and see if we have a balance. Note genesis block transactions have a maturity time of 100 blocks, so we must mine 100 blocks as well. Let's convert the private key from chapter 3.2 to Wallet Import Format (WIF) to import to the Bitcoin Core wallet. Note it is an uncompressed WIF (thus –u flag) because it was an uncompressed public key:

```
$ wif 2358feea3003a1d16af3454be4cec2f6a7db43bfc7daa101b8949fff91ed64b4 -n regtest -u
91rV55q8XVkFSY6rqfGbR2RAGfdfZDmdVycD4b9Z3CffXWkEf5w
$ learncoin-cli -regtest importprivkey 91rV55q8XVkFSY6rqfGbR2RAGfdfZDmdVycD4b9Z3CffXWkEf5w premine
$ learncoin-cli -regtest listlabels
[
    "",
    "premine"
]
$ learncoin-cli -regtest getbalance
0.00000000
```

3 of 4 2/5/19, 10:02 PM

```
$ learncoin-cli -regtest generate 100
[
 "..."
 ...
]
$ learncoin-cli -regtest getbalance
1000000.000000000
```

Transaction

Let's create another address that we can test sending coins to:

```
$ learncoin-cli -regtest getnewaddress
t764XjzmL1tYztdccMHwHzLEmGoKFTsf2w
$ learncoin-cli -regtest sendtoaddress t764XjzmL1tYztdccMHwHzLEmGoKFTsf2w 100000
...(txid)...
$ learncoin-cli -regtest getbalance
899999.99996260
```

You can see that it deducted the amount (plus a small transaction fee) for the transaction.

4 of 4