

Using the hypernode ledger: quick start guide

start your server

```
-MacBook-Pro ledger2 % java -jar /Users/shinji/Downloads/ledger/target/ledger-0.0.1-SNAPSHOT.jar
```

if you don't already have a key pair get it on here:

Paste JSON here:

```
xSSgZkaLXjcrvUu3c7+ctJ3RtcM5ZLBnttVUiG9VbtZuh5XEJu  
NaUiD//e/5yBH6ww3h4+5P0VX2DTIm5iZJkYtPaqg7FU/  
nn1+LuLuWa3QECJBQz674iiriMj0b/  
l18ZoP+0iuBu8lhY2oJcLMXdhsWo620dQ1ALyZEaf/Fy1d/  
jpls3udZB0r2hnQNzP0p7cMQgB3siReDTUc7hziCjELZQ/  
Z8ZDppV0St912byD1AW0tpG9yu8uGT32zIYUgypQzRHL0oCaU/  
A5Bfv1MYHL8f8MgUwLFRm0m9mh0YlpcLwe03em8W4SDYgrW3F1  
A3AvyjQP6mkJf4xXnBk7Z0aNlx+aPCl72w979jrN1daZVmz2PC  
C8RcwFlUohR7tfi/zzPLm0JLaF2IDbdDDB9xVAgMBAAE=  
}
```

Send

OK

now set the desired keys into the server

Paste JSON here:

```
xSSgZkaLXjcrvUu3c7+ctJ3RtcM5ZLBnttVUiG9VbtZuh5XEJu  
NaUiD//e/5yBH6ww3h4+5P0VX2DTIm5iZJkYtPaqg7FU/  
nn1+LuLuWa3QECJBQz674iiriMj0b/  
l18ZoP+0iuBu8lhY2oJcLMXdhsWo620dQ1ALyZEaf/Fy1d/  
jpls3udZB0r2hnQNzP0p7cMQgB3siReDTUc7hziCjELZQ/  
Z8ZDppV0St912byD1AW0tpG9yu8uGT32zIYUgypQzRHL0oCaU/  
A5Bfv1MYHL8f8MgUwLFRm0m9mh0YlpcLwe03em8W4SDYgrW3F1  
A3AvyjQP6mkJf4xXnBk7Z0aNlx+aPCl72w979jrN1daZVmz2PC  
C8RcwFlUohR7tfi/zzPLm0JLaF2IDbdDDB9xVAgMBAAE=  
}
```

Send

OK

time to create the new ledger (remember to specify account name and connection string, as this is the connection string other servers will use to connect to you)

```
{  
  "id": 1,  
  "distributedLedgerAccounts": [  
    {  
      "publicKey": "MIIBoqANBgkqhkiG9w0BAQEFAACAY8AMIIBigKCAYEAvnHa4H05W7iHobIGp6X9N6eC2MfTP60VAPvtzshapfgiGbdGBqfViXnfRtZ1lWAjInvaquU9wPgaJQfx06M8v/  
EAj2GnCEfZ+lxu+91lx/XSgZkaLxjcrvUu3c7+c73rtcm5ZLBnttVUiG9VbtZuh5XEjuNaUiId//e/5yBH6ww3h4+5P0VX2DTIm5iZjKytPaqq7FU/n+1+LuLuWa30ECJBQz674iirimj0b/  
l18ZoP+0iuBu8lhYzojCLMxdhSw0620dQ1ALyZEaf/Fy1d/jpls3udZB0r2hnQNzP0p7cM0gB3s1reDTUc7hziCjELZQ/Z8ZDppV0St912byD1AWotp69yu8uGT32zIYuyp0zRHLooCaU/  
A5Bfv1MYHL8f8MgUwLFrM0m9mh0YlpcLwe03em8W4SDYgrW3F1A3AvyjQP6mkJf4xXnBk7Z0aNLX+aPC172w979jrn1daZVmz2PCC8RcwFlUohR7tfi/zzPLm0JLaF2IDbdDB9xVAgMBAAE=",  
      "name": "ACCOUNT1_NAME",  
      "amount": "1000.00",  
      "publicKey": "MIIBoqANBgkqhkiG9w0BAQEFAACAY8AMIIBigKCAYEAvnHa4H05W7iHobIGp6X9N6eC2MfTP60VAPvtzshapfgiGbdGBqfViXnfRtZ1lWAjInvaquU9wPgaJQfx06M8v/  
EAj2GnCEfZ+lxu+91lx/XSgZkaLxjcrvUu3c7+c73rtcm5ZLBnttVUiG9VbtZuh5XEjuNaUiId//e/5yBH6ww3h4+5P0VX2DTIm5iZjKytPaqq7FU/n+1+LuLuWa30ECJBQz674iirimj0b/  
l18ZoP+0iuBu8lhYzojCLMxdhSw0620dQ1ALyZEaf/Fy1d/jpls3udZB0r2hnQNzP0p7cM0gB3s1reDTUc7hziCjELZQ/Z8ZDppV0St912byD1AWotp69yu8uGT32zIYuyp0zRHLooCaU/  
A5Bfv1MYHL8f8MgUwLFrM0m9mh0YlpcLwe03em8W4SDYgrW3F1A3AvyjQP6mkJf4xXnBk7Z0aNLX+aPC172w979jrn1daZVmz2PCC8RcwFlUohR7tfi/zzPLm0JLaF2IDbdDB9xVAgMBAAE=",  
      "validatorNodeList": [  
        {"connectionString": "CONNECTION_STRING",  
         "publicKey": "MIIBoqANBgkqhkiG9w0BAQEFAACAY8AMIIBigKCAYEAvnHa4H05W7iHobIGp6X9N6eC2MfTP60VAPvtzshapfgiGbdGBqfViXnfRtZ1lWAjInvaquU9wPgaJQfx06M8v/  
EAj2GnCEfZ+lxu+91lx/XSgZkaLxjcrvUu3c7+c73rtcm5ZLBnttVUiG9VbtZuh5XEjuNaUiId//e/5yBH6ww3h4+5P0VX2DTIm5iZjKytPaqq7FU/n+1+LuLuWa30ECJBQz674iirimj0b/  
l18ZoP+0iuBu8lhYzojCLMxdhSw0620dQ1ALyZEaf/Fy1d/jpls3udZB0r2hnQNzP0p7cM0gB3s1reDTUc7hziCjELZQ/Z8ZDppV0St912byD1AWotp69yu8uGT32zIYuyp0zRHLooCaU/  
A5Bfv1MYHL8f8MgUwLFrM0m9mh0YlpcLwe03em8W4SDYgrW3F1A3AvyjQP6mkJf4xXnBk7Z0aNLX+aPC172w979jrn1daZVmz2PCC8RcwFlUohR7tfi/zzPLm0JLaF2IDbdDB9xVAgMBAAE="],  
        "ledgerParameters": {  
          "messageUpdateFrequencyPerHour": 30,  
          "frameProcessingTimeMilliseconds": 5000,  
          "maxConnections": 20,  
          "amountRequestedToBeValidator": "1",  
          "transactionCost": "0.1",  
          "maxTransactionsPerBlock": 1000,  
          "transmitRedundancy": 1  
        }  
      }  
    }  
  ]  
}
```

Send

now your system is online.

You can specify other public keys in the new ledger, or if you want to add new people to the system you can make a payment

This is how to make a payment: first configure the client by entering your private and public key

Set Encryption Entity (Integrated)

```
{  
  "privateKey": "  
MIIG/  
wIBADANBgkqhkiG9w0BAQEFAASCBukwggblAgEAAoIBgQC+cdr  
gdLbuIehsganpf03p4LYx9M/rRUA+  
+30yFql+CIzt0YGp9UjGd9G1nWVYCMg29qq5T3A+BqNB/  
HTozy/8QCPYY1wR9n6XG772KXH/  
FJKBmRoteNyu9S7dzv5y0ndG1wzlksGe21VSIb1Vu1m6HlcQm4  
1pQh3/97/nIEfrDDeHj7k/RVfYNMibmJkmRi09qqDsVT+ef/  
X4u4u5ZrdAQIkFDPrviKKuIyM5v+XXxmng/46K4G7yWFjaglwSX",  
  "publicKey": "MIIBOQIDAQABAAQDwX...  
  "alias": "MyEncryptionEntity"  
}
```

Send

Already Started

and then set the target server (in this case I'm using localhost but you will likely use a different server with a proper web address unless you're on the same machine)

Paste validator connection string here:

```
http://localhost:8080
```

Send

OK

now you can make a payment. If the recipient exists already you can use the name in the to field, otherwise you will have to use the recipient's public key.

Paste JSON here:

```
{
  "to": "Node5",
  "comment": "COMMENT22",
  "amount": "22.00"
}
```

Send

To change the account parameters of an account(you need to follow the steps to be the authenticated in the client just as you did to pay)

Paste JSON here:

```
{
  "from": "ACCOUNT_PUBLIC_KEY",
  "name": "ACCOUNT_NEW_NAME",
  "delegated": "PUBLIC_KEY_SERVER"
}
```

Send

To connect as a server into an existing ledger:



Set Encryption Entity (Integrated)

Paste JSON here:

```
{  
  "privateKey":  
    "MIIG/  
AIBADANBgkqhkiG9w0BAQEFAASCBuYwggbiAgEAAoIBgQC3h9F  
tta0KuBrj7x4JvYncHT0x9Y589ZfBZlg9nLpS3Hwrs1mdwP9S  
YsmxHqlfl3ZsirBzbQFgfM4KLvdT4zThyM/  
qTnCOG2HcBo3KJUwkKUuw18601rdsK645A0gbPJfftSiKHE6/  
ZigRf19E0owX7j.cj fDhN+glTp4gkLGYfHoHp4Gi.TRElQbn5Rxk  
3Decy2Z0pw86G5NxslkxySLitdS59ULu4gCL024beQuv9xbau  
3fud0AUrzEkzoV8BijgVjcAaDMaTv/4LzTXcLbFmHp+RzN3UY#
```

Send

then join the existing ledger (use the string others will use to connect to you, I'm using localhost as I'm testing locally but to use it for real you will have to specify reachable addresses)

It will take a while for your request to be processed, so watch the transmitted message of the connectionstringdestination to monitor its status (it will take 2 full block ids to complete)

The screenshot shows a JSON response from the /getStatus endpoint. The response includes fields like id, signature, publicKey, messageValue, and distributedLedgerAccounts. The distributedLedgerAccounts field contains two entries, each representing a ledger account with fields like name, amount, and validatorNode.

```
{  
  "id": 6,  
  "signature":  
    "publicKey": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAvnHa4H05W7iHobIGp6X9N6eC2MfTP60VAPu...  
    "messageValue": null,  
    "signatureValue": "PuVusipi42Jkw25Rd2KE+Lyg21/F6cxw4BfdPXxppTkHX/NFlfvYilvkShaMKj77+/0BaLKZrZCdV5...  
  "distributedLedgerAccounts":  
    "0":  
      "publicKey": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAt4fRbbWpTirga4+8eCb2J3B0zsfW0fpWxw...  
      "name": "Node1",  
      "amount": "1000.00 [JS: 1000]",  
      "validatorNode": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAt4fRbbWpTirga4+8eCb2J3B0zsfW0fpWxw...  
    "1":  
      "publicKey": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAt4fRbbWpTirga4+8eCb2J3B0zsfW0fpWxw...  
      "name": "Node2",  
      "amount": "1000.00 [JS: 1000]",  
      "validatorNode": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAt4fRbbWpTirga4+8eCb2J3B0zsfW0fpWxw..."  
}
```

you can use the getStatus to verify that the server is running correctly and you can see the transmitted message

The screenshot shows a JSON response from the /getCurrentlyTransmittedTransportMessage endpoint. It includes fields like blockId, blockRevision, blockTempVersion, hash, signature, and signedValidatorMessages. The signedValidatorMessages field contains one entry, which includes a contract and votedParameterChanges.

```
{  
  "blockId": 7,  
  "blockRevision": 3,  
  "blockTempVersion": 2,  
  "hash": "7f843282efc8ced743fb45835822f5d",  
  "signature":  
    "publicKey": "StarterNode",  
    "messageValue": null,  
    "signatureValue": "8xwrxt1dLuulxB0B7i82N20gU0PAk7vt495tcpUln14x54501rXAJvcAa/0GnIsG/Zwvmou6RH46ITCIHYRZuriLf1rhwLvxu8Gf+VyvrerDIAGFnz1vJtbux9+6xPN0n+QQGm56wamgjq/lyTAOT0Pz4D7WByeDKu0GVRJrhk8wbp9mtnn0+2jb2zg4AlHweWzAdrpsks5s1ZM",  
  "signedValidatorMessages":  
    "0":  
      "contract":  
        "id": 7,  
        "validatorNodes": [],  
        "paymentSet": [],  
        "votedParameterChanges":  
          "maxMessageLength": 0  
}
```

to vote for different parameters or to reassign the values of a particular account you can vote to do so by calling /changeVotedParameters. It will need the majority of votes to agree on a changeset before it is applied

Paste JSON here:

```
{ "messageUpdateFrequencyPerHour": 30, "frameProcessingTimeMilliseconds": 5000, "maxConnections": 20, "amountRequestedToBeValidator": "1", "transactionCost": "0.1", "maxTransactionsPerBlock": 1000, "distributedLedgerAccountReassignProposals": [ { "publicKey": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAunH4nCceRoBkLW1Ec0j4plIxBnUVH6LcPRk2s5wG0rW8SP7gdMHH2wywh7fVT8GWo0VLr/9Wz1kyS8Gwm3X0TmLX0YEyz1ogHSy4Sp2cDls5XMX9ct4SssMoz6EU+tkhCNHyM0KUd8hvynrT7YTbCxSwJKmz+Wez1K93wb+jV6pi4tH9x0+4fyxq38Z3/MK7U/D00ne0CMQGB7KH0qUrTG6ZEB57X4h1fr0mmwvBqnQiGDDdsVf+yHJEvbPJjoCshg8l303TxqZutzEup7D2tPDKkm8xmRGl1FqoWaHoa86urW1+J0rIpaUhujLZwgGUukV6yyF47MdKV4PbQMDmbHAXhLM81wbnv7nYpNIowTA930XjHPfkH2EkCP2a7U/qz9H60e7lKyd9oxkbcdXxU2+bbt094mgBPsIewhspBuUWygqXeJo8ByI73j6ynow0vIc7v8EHRuLPgAZl2/zmXuwIChYf46w8auNeMV68aNll+Au11LE5SGy3kHRAgMBAAE=", "name": "SHINJT", "amount": "1000.00", }, { "validatorNode": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAunH4nCceRoBkLW1Ec0j4plIxBnUVH6LcPRk2s5wG0rW8SP7gdMHH2wywh7fVT8GWo0VLr/9Wz1kyS8Gwm3X0TmLX0YEyz1ogHSy4Sp2cDls5XMX9ct4SssMoz6EU+tkhCNHyM0KUd8hvynrT7YTbCxSwJKmz+Wez1K93wb+jV6pi4tH9x0+4fyxq38Z3/MK7U/D00ne0CMQGB7KH0qUrTG6ZEB57X4h1fr0mmwvBqnQiGDDdsVf+yHJEvbPJjoCshg8l303TxqZutzEup7D2tPDKkm8xmRGl1FqoWaHoa86urW1+J0rIpaUhujLZwgGUukV6yyF47MdKV4PbQMDmbHAXhLM81wbnv7nYpNIowTA930XjHPfkH2EkCP2a7U/qz9H60e7lKyd9oxkbcdXxU2+bbt094mgBPsIewhspBuUWygqXeJo8ByI73j6ynow0vIc7v8EHRuLPgAZl2/zmXuwIChYf46w8auNeMV68aNll+Au11LE5SGy3kHRAgMBAAE=", "publicKey": "MIIBojANBgkqhkiG9w0BAQEFAAOCAy8AMIIIBigKCAYEAk5GJ451dsBR9rPqxDmWE0ab/FVIYItP208JHjCe6Pfvf+73vJuXIRLs+yZxohAp+1XNNTM4CKFLB+LSC+KUTPDU+2XV1Q9IP+K019+4+MSBEE+GTZ+1GCHVFG+O4" } ] }
```

Send

this is a very basic user interface, a GUI expert could surely improve on this and they're encouraged to do so and publish their changes under the AGPL license, so that the whole community can benefit.

To send public and private keys to someone else is very bad practice, so always run your own client on your own device and never connect to someone else's client that is not open source and not running directly on devices you control