**Setting up Hyperledger Explorer using First Network Example from Fabric Samples**

**Description:**

In this Tutorial we setup the „Fabric Samples – First Network“ and the „Hyperledger Explorer“ for a better depiction.

**Procedure:**

1. Install prerequisites
2. Clone the Hyperledger Explorer Repository
3. Setup the Database using Postgres
4. Clone the Hyperledger Fabric Samples and setup the network
5. Adjust Hyperledger Explorer to Fabric Samples - First Network
6. Build Hyperledger Explorer
7. Run Hyperledger Explorer
8. Check out the Hyperledger Explorer GUI

**Implementation:**

1. Install prerequisites

Now we setup all the necessary requirements for running the Hyperledger Explorer.

**Nodejs 6.x** (Nodejs Version 7.x is not supported yet)

* Pull the latest Version of Nodejs 6.x:

***$ curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -***

* Install it using:

***$ sudo apt-get install -y nodejs***

**Docker CE**

* Update the apt package index:

***$ sudo apt-get update***

* Install packages to allow apt to use the repository:

***$ sudo apt-get install \***

***apt-transport-https \***

***ca-certificates \***

***curl \***

***software-properties-common***

* Add Dockers official GPG Key:

***$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –***

* Setup the Stable Repository:

***$ sudo add-apt-repository \***

***"deb [arch=amd64] https://download.docker.com/linux/ubuntu \***

***$(lsb\_release -cs) \***

***stable"***

* Update apt package index again:

***$ sudo apt-get update***

* Install the latest version of Docker CE:

***$ sudo apt-get install docker-ce***

**Docker-Compose**

* Download the latest version of Docker-Compose:

***$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker- compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose***

Apply executable permissions to the binary:

***$ sudo chmod +x /usr/local/bin/docker-compose***

**PostgreSQL**

Update apt package index:

***$ sudo apt-get update***

Install PostgreSQL via the apt package index:

***$ sudo apt-get install postgresql postgresql-contrib***

2. Clone the Hyperledger Explorer Repository

Now we will clone the latest repository of Hyperledger Explorer.

* Clone the repository:

***$ git clone https://github.com/hyperledger/blockchain-explorer.git***

3. Setup the Database using Postgres

Now we will setup and connect to the PostgreSQL database.

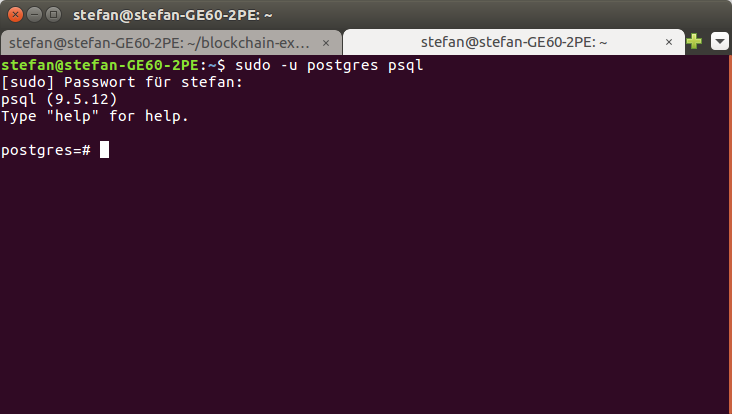
* First we change to the directory of Hyperledger Explorer

***$ cd blockchain-explorer***

* Now we connect to the PostgreSQL database

***$ sudo -u postgres psql***

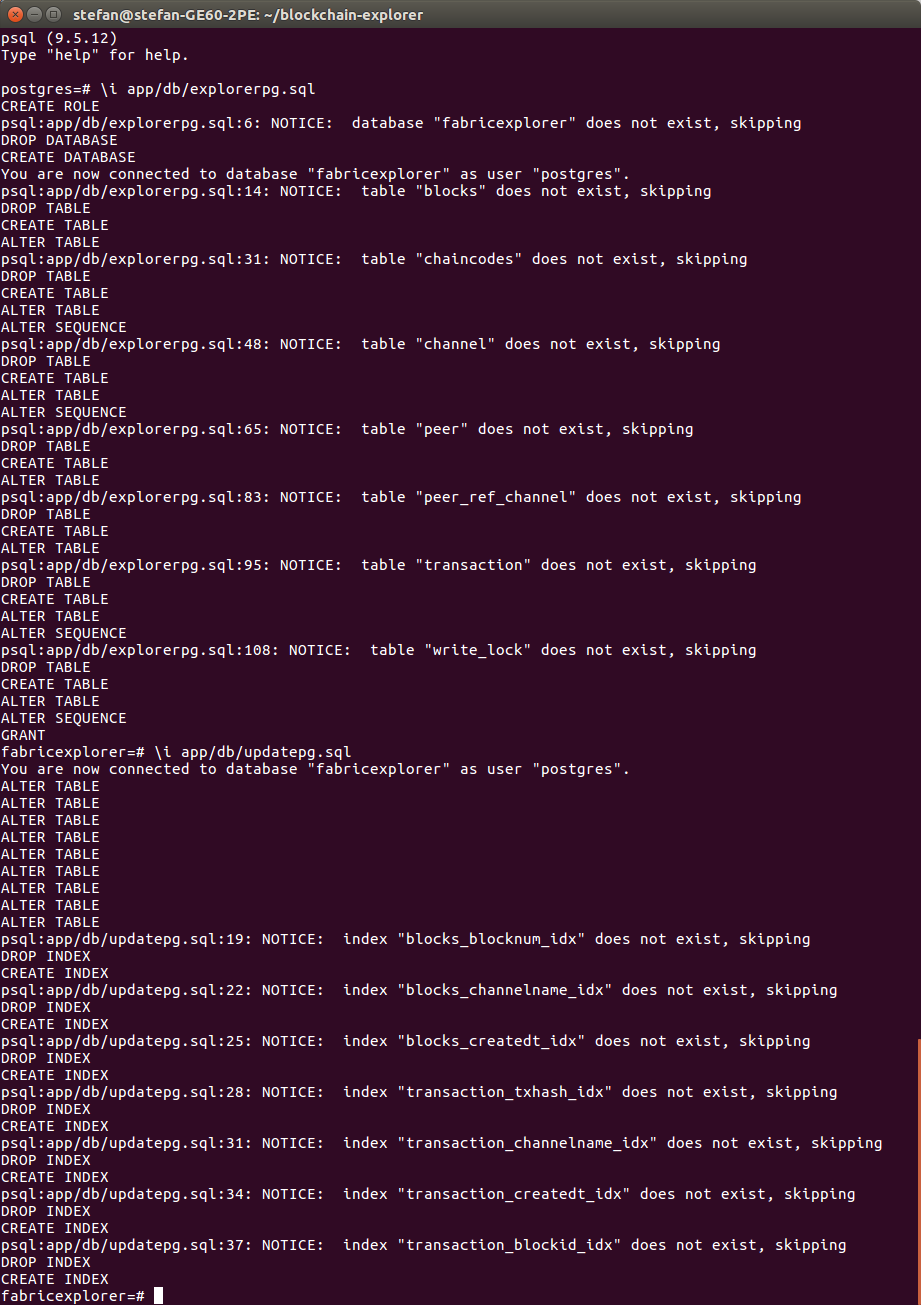
You should see following:



* Next we will create the database running following scripts inside postgres:

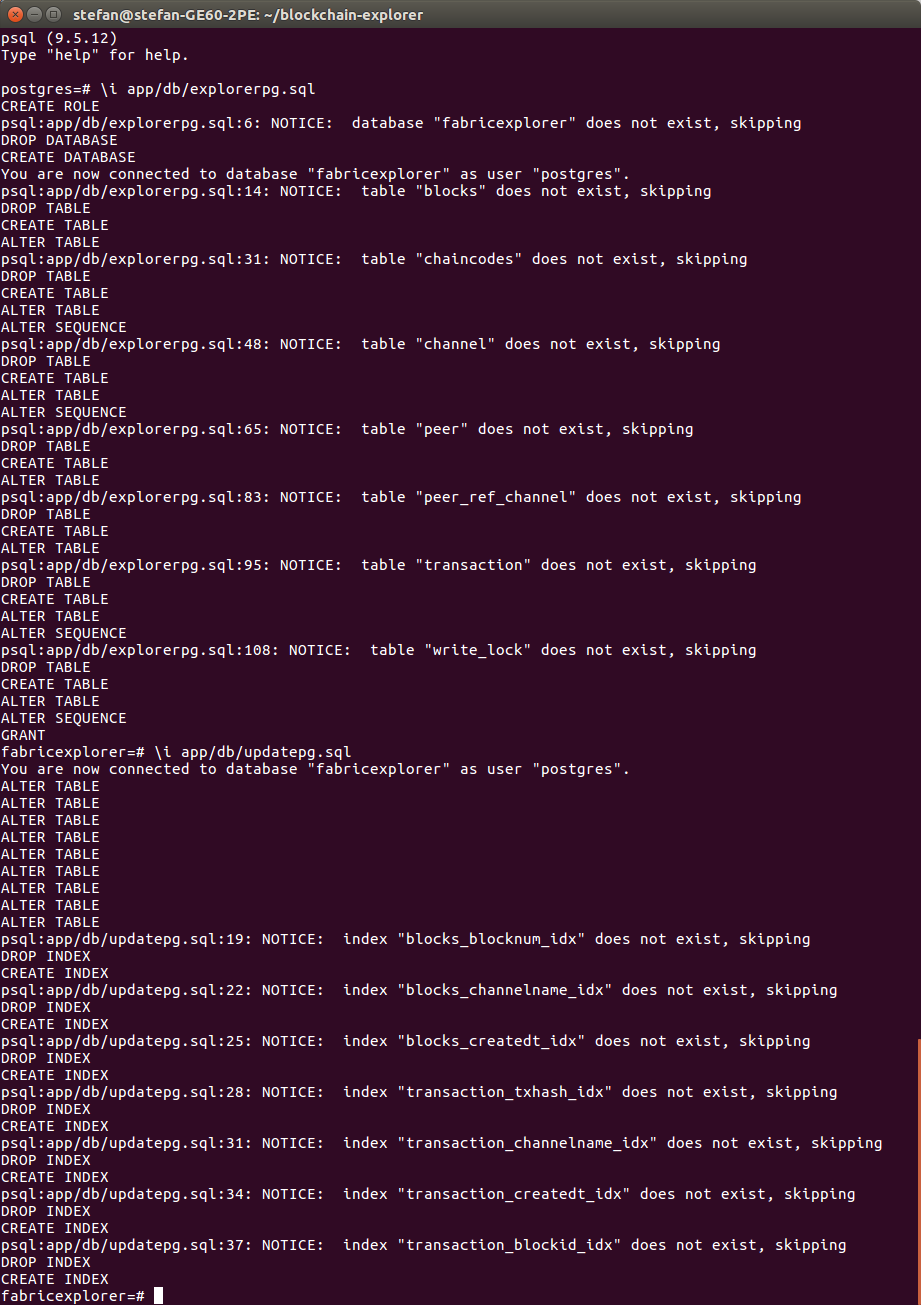
***$ \i app/db/explorerpg.sql***

You should see following:



* Next we will excute the second script:

***$ \i app/db/updatepg.sql***



4. Clone the Hyperledger Fabric Samples and setup the network

Now we will clone the Hyperledger Fabric Samples, checkout with the Version v1.0.6 and setup the network using the first-network tutorial. We were using the Hyperledger Fabric v1.0 because of compatibility reasons.

* Open another terminal
* First we clone the fabric-samples repository and change inside the fabric samples directory:

***$ git clone https://github.com/hyperledger/fabric-samples.git***

***$ cd fabric-samples/***

* Next we checkout with the version v1.0.6

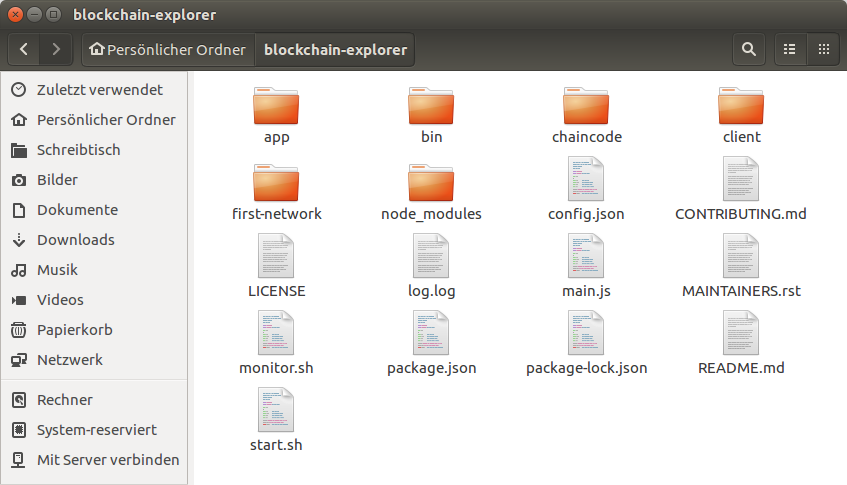
***$ git checkout v1.0.6***

In order to make it more clear for us we copied the folders „chaincode“ and „first-network“, which are necessary for the network setup, into the blockchain-explorer directory.

* Now pull and extract the necessary platform-specific binaries and Docker images with the Version 1.0.6:

***$ curl -sSL https://goo.gl/kFFqh5 | bash -s 1.0.6***

It should look similar like in the picture below:



* Now we have to into the first network directory:

***$ cd blockchain-explorer/first-network/***

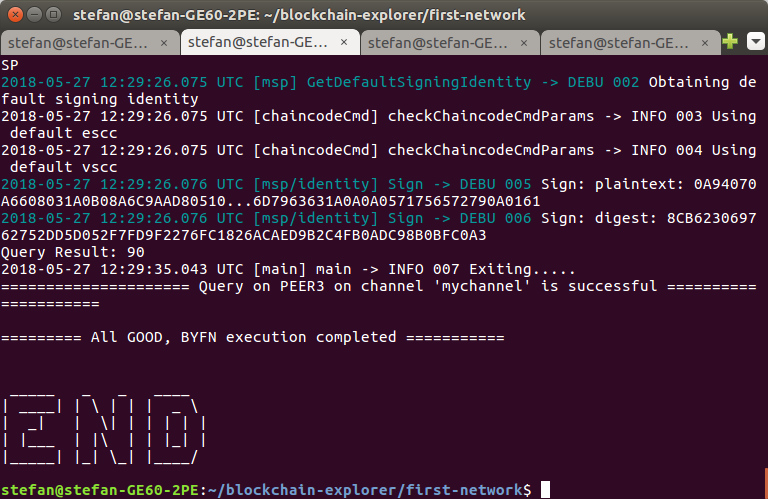
* Now we generate all necessary Certificates and Network Artifacts with the provided script byfn.sh in order to run the network:

***$ ./byfn.sh -m generate***

* Now we bring up the network with the provided script byfn.sh:

***$ ./byfn.sh -m up***

If you see the END sign in your terminal, you know the script executed sucessfully and that your network is up running!



5. Adjust Hyperledger Explorer to Fabric Samples - First Network

Now we will adjust the Hyperleder Explorer to the Network of the Fabric Samples by modifying the config.json file located in the blockchain-explorer/ directory.

Original File config.json

{

"network-config": {

"org1": {

"name": "peerOrg1",

"mspid": "Org1MSP",

"peer1": {

"requests": "grpcs://127.0.0.1:7051",

"events": "grpcs://127.0.0.1:7053",

"server-hostname": "peer0.org1.example.com",

"tls\_cacerts": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"

},

"peer2": {

"requests": "grpcs://127.0.0.1:8051",

"events": "grpcs://127.0.0.1:8053",

"server-hostname": "peer1.org1.example.com",

"tls\_cacerts": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org1.example.com/peers/peer1.org1.example.com/tls/ca.crt"

},

"admin": {

"key": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/keystore",

"cert": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/signcerts"

}

},

"org2": {

"name": "peerOrg2",

"mspid": "Org2MSP",

"peer1": {

"requests": "grpcs://127.0.0.1:9051",

"events": "grpcs://127.0.0.1:9053",

"server-hostname": "peer0.org2.example.com",

"tls\_cacerts": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt"

},

"peer2": {

"requests": "grpcs://127.0.0.1:10051",

"events": "grpcs://127.0.0.1:10053",

"server-hostname": "peer1.org2.example.com",

"tls\_cacerts": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org2.example.com/peers/peer1.org2.example.com/tls/ca.crt"

},

"admin": {

"key": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp/keystore",

"cert": "fabric-path/fabric-samples/first-network/crypto-config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp/signcerts"

}

}

},

"host": "localhost",

"port": "8080",

"channel": "mychannel",

"keyValueStore": "/tmp/fabric-client-kvs",

"eventWaitTime": "30000",

"users":[

{

"username":"admin",

"secret":"adminpw"

}

],

"pg": {

"host": "127.0.0.1",

"port": "5432",

"database": "fabricexplorer",

"username": "hppoc",

"passwd": "password"

},

"license": "Apache-2.0"

}

End of Original File config.json

Modifications are highlighted using the **bold** function. You need to adjust the path of „tls\_cacerts“ of each peer in the organisations and you need to adjust all the key and cert paths of the admin of each organisation. Instead of user you are supposed to write the username of your machine.

Modified File config.json

{

"network-config": {

"org1": {

"name": "peerOrg1",

"mspid": "Org1MSP",

"peer1": {

"requests": "grpcs://127.0.0.1:7051",

"events": "grpcs://127.0.0.1:7053",

"server-hostname": "peer0.org1.example.com",

"tls\_cacerts": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt"

},

"peer2": {

"requests": "grpcs://127.0.0.1:8051",

"events": "grpcs://127.0.0.1:8053",

"server-hostname": "peer1.org1.example.com",

"tls\_cacerts": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org1.example.com/peers/peer1.org1.example.com/tls/ca.crt"

},

"admin": {

"key": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/keystore",

"cert": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp/signcerts"

}

},

"org2": {

"name": "peerOrg2",

"mspid": "Org2MSP",

"peer1": {

"requests": "grpcs://127.0.0.1:9051",

"events": "grpcs://127.0.0.1:9053",

"server-hostname": "peer0.org2.example.com",

"tls\_cacerts": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt"

},

"peer2": {

"requests": "grpcs://127.0.0.1:10051",

"events": "grpcs://127.0.0.1:10053",

"server-hostname": "peer1.org2.example.com",

"tls\_cacerts": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org2.example.com/peers/peer1.org2.example.com/tls/ca.crt"

},

"admin": {

"key": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp/keystore",

"cert": "**/home/USER/blockchain-explorer**/first-network/crypto-config/peerOrganizations/org2.example.com/users/Admin@org2.example.com/msp/signcerts"

}

}

},

"host": "localhost",

"port": "8080",

"channel": "mychannel",

"pg": {

"host": "127.0.0.1",

"port": "5432",

"database": "fabricexplorer",

"username": "hppoc",

"passwd": "password"

},

"keyValueStore": "/tmp/fabric-client-kvs",

"eventWaitTime": "30000",

"users":[

{

"username":"admin",

"secret":"adminpw"

}

],

"license": "Apache-2.0"

}

End of Modified File config.json

6. Build Hyperledger Explorer

Now we build up the Hyperledger Explorer using the package manager npm.

* Run the following command in order:

***$ cd blockchain-explorer/app/***

***$ npm install***

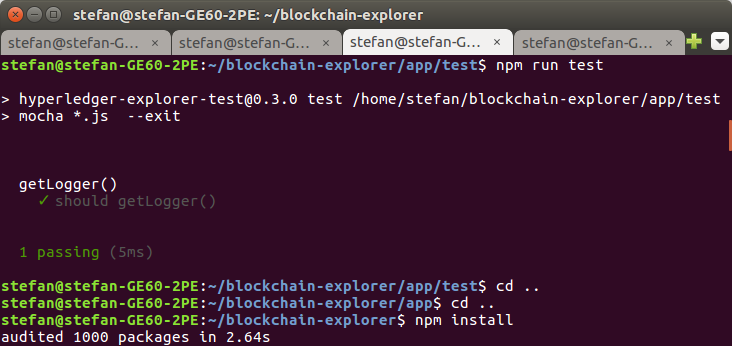
***$ npm audit fix***

***$ cd test/***

***$ npm install***

***$ npm run test***

You should see following:



***$ cd ..*** (2 times so you get back to the „blockchain-explorer/“ directory)

***$ npm install***

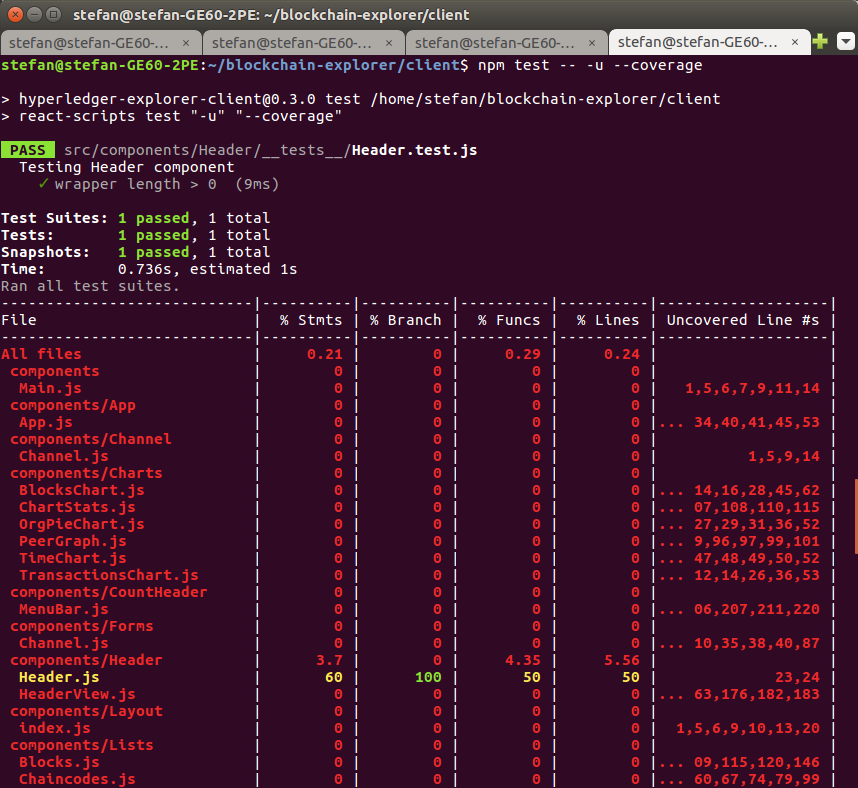
***$ npm audit fix***

***$ cd client/***

***$ npm install***

***$ npm test -- -u –coverage***

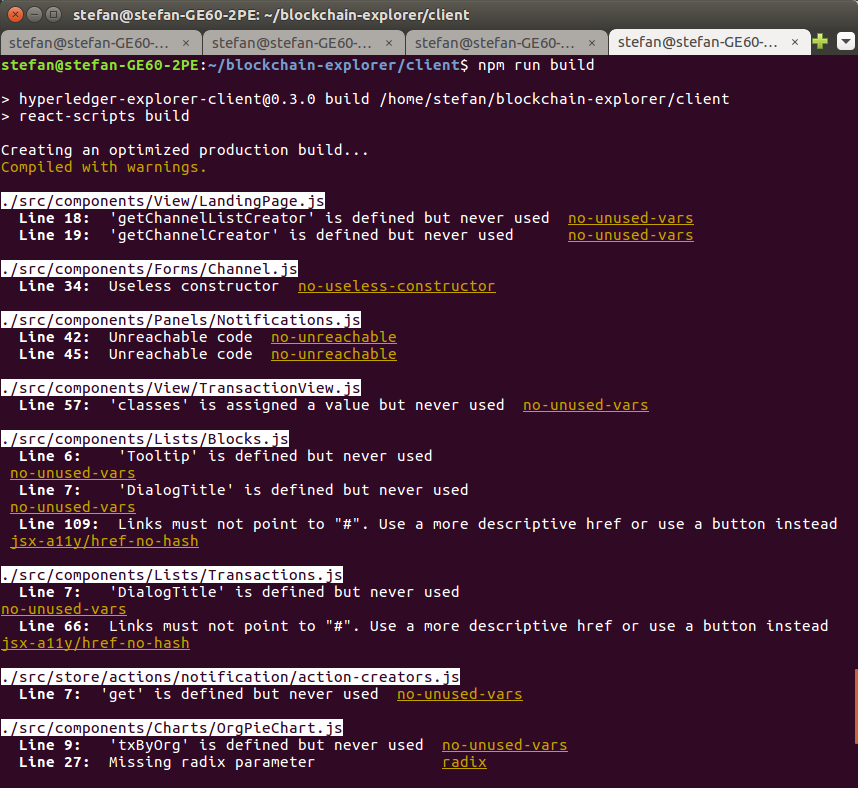
You should see following:



* + Last command:

***$ npm run build***

You should see following:



Now Hyperledger Explorer is build up and ready to run.

7. Run Hyperledger Explorer

Now we will make Hyperledger Explorer run by executing the given script.

* + You have to be in the blockchain-explorer directory and execute the start.sh script to run the Explorer:

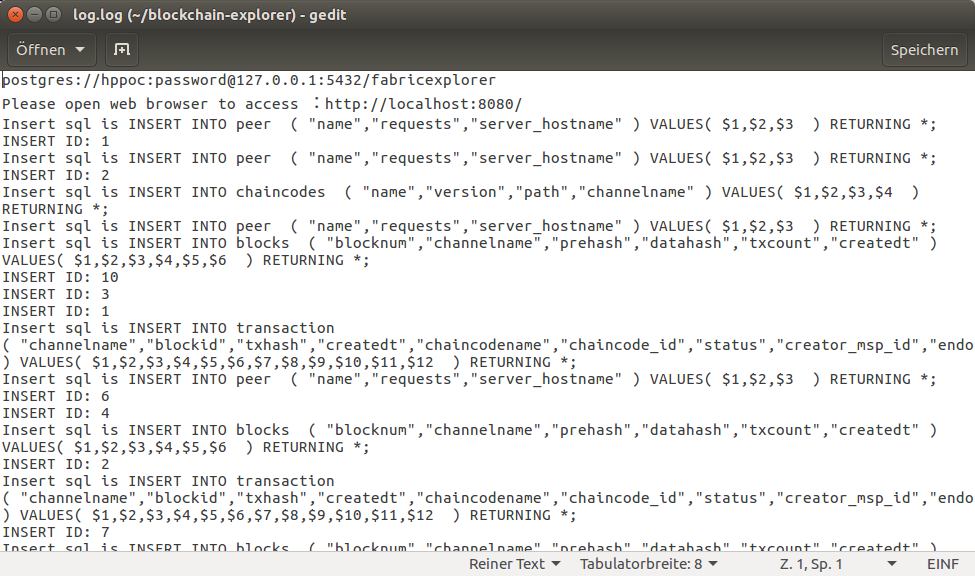
***$ ./start.sh***

* + Check logs for mistakes:

***$ tail -f log.log***

After executing the script open the created log.log file located in the directory which you are in right now.

If everything worked out, it should look like this:



If you have any error, the Hyperledger Explorer is not running probably!

**Troubleshooting:**

A path to tls\_cacert or admin key and cert could be wrong.

If an Inotify-Resource error occures fix it with following command:

***$ sudo su***

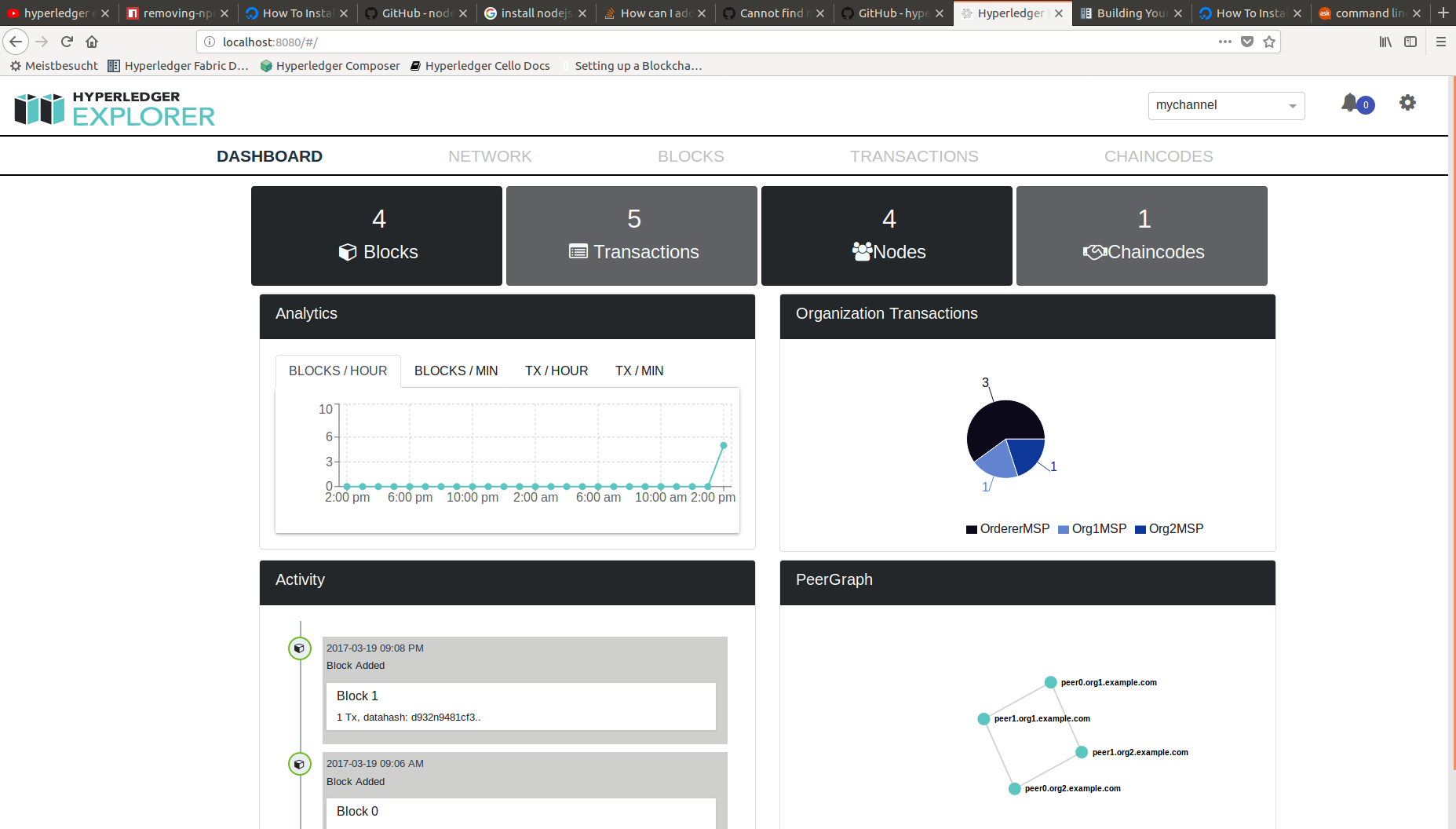
***$ echo 65535 > /proc/sys/fs/inotify/max\_user\_watches***

***$ exit***

In case of any error your have to delete the folders node\_modules in every folder where we ran the command: $npm install and start from 6. Building Hyperledger Explorer again.

8. Check out the Hyperledger Explorer GUI

If everything worked out as expected, your will be able to launch following URL [http://localhost:8080](http://localhost:8080/) on a browser and Hyperledger Explorer will be shown:



Resources used in the Tutorial

- [https://docs.docker.com/install/linux/docker-ce/ubuntu/#set-up-the-repository](https://docs.docker.com/install/linux/docker-ce/ubuntu/" \l "set-up-the-repository)

- [https://docs.docker.com/compose/install/#install-compose](https://docs.docker.com/compose/install/" \l "install-compose)

- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-16-04>

- <https://github.com/hyperledger/blockchain-explorer>

- <https://hyperledger-fabric.readthedocs.io/en/release-1.0/build_network.html>