QR Code Server

Operating System: Ubuntu

AWS or local installation

The Verification, QR Code, and Caching server can either be all installed on separate computers, or all run on the same computer. These instructions will outline how to set up the QR Code server as a separate server, and at the end there will be instructions on how to modify the server if you want to run them all on the same computer for simplicity.

Each Node.js server requires `npm` be installed, as well as all the dependencies for the project.

Install npm

The first step is to install npm so that the Node.js servers can run. Run the following command in a new terminal

```
sudo apt-get install npm
```

We now need to download the web applications from GitHub. Perform the following command in a terminal in your `home` directory.

```
git clone https://github.com/SwinburneBlockchain/WebApplications
```

This copies the Node.js web server from the GitHub link. If you plan on running each Node.js server on individual servers then you may delete the `VerificationServer.js` and `CachingServer.js` files that were downloaded along with the `QRCodeServer.js` file. Otherwise continue onto the next step.

To initialise this application, you will need to move into the WebApplications folder created when you cloned the GitHub link, and initialise npm in this folder.

```
cd WebApplications
npm init
```

You will now need to edit the `package.json` file that was downloaded from GitHub. Open up the `package.json` file, and delete the following line:

```
"mongodb": "^3.2.16"
```

Now follow these instructions on installing MongoDB on Ubuntu

https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-16-04

If you are having trouble starting MongoDB after following these instructions, see Appendix 1.

You should now be able to run the following command which installs the remaining dependencies for the web server:

```
npm install
```

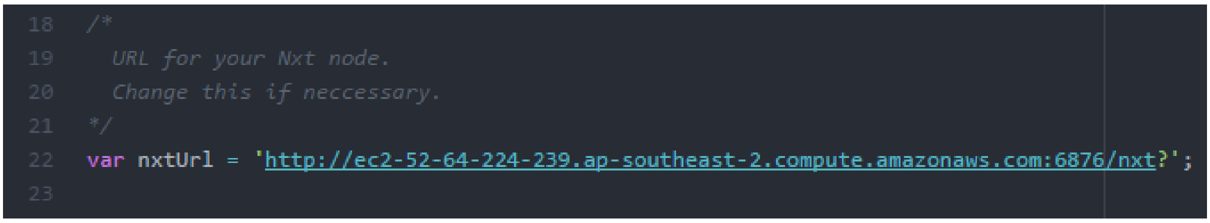
Once everything has been installed, run the Verification Server by entering the following command:

```
node QRCodeServer.js
```

Modifying Program Constants

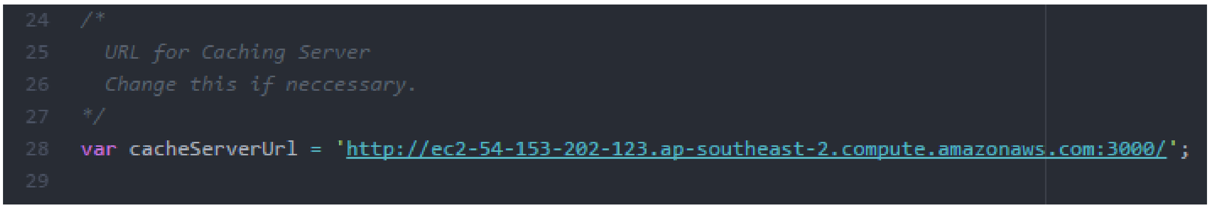
There are several values in `QRCodeServer.js` that you will need to modify to suit your setup.

The first is the address of your online Nxt node, on line 22. Change this to be the same address as the node you created in “Set up Nxt Nodes” documentation. You will need to add `:6876/nxt?` to the end of your URL.



```
18  /*
19    URL for your Nxt node.
20    Change this if neccessary.
21  */
22  var nxtUrl = 'http://ec2-52-64-224-239.ap-southeast-2.compute.amazonaws.com:6876/nxt?';
23
```

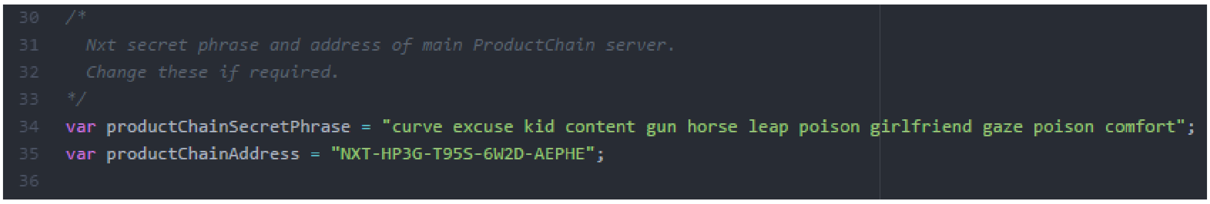
The second is the URL (or IP address) of your Caching Server on line 28. Remember to add port 3000 to the end of your URL (the port the Node.js server is running on)



```
24  /*
25    URL for Caching Server
26    Change this if neccessary.
27  */
28  var cacheServerUrl = 'http://ec2-54-153-202-123.ap-southeast-2.compute.amazonaws.com:3000/';
29
```

You will also need to put in the secret phrase and Nxt address of your ProductChain account on lines 34 and 35. This is the account you generated in “Set up Nxt Nodes” documentation. Alternatively you can use our existing account which already holds Nxt for you to use.

```
Nxt Address:     NXT-HP3G-T95S-6W2D-AEPHE
Secret Phrase:   curve excuse kid content gun horse leap poison
                 girlfriend gaze poison comfort
```

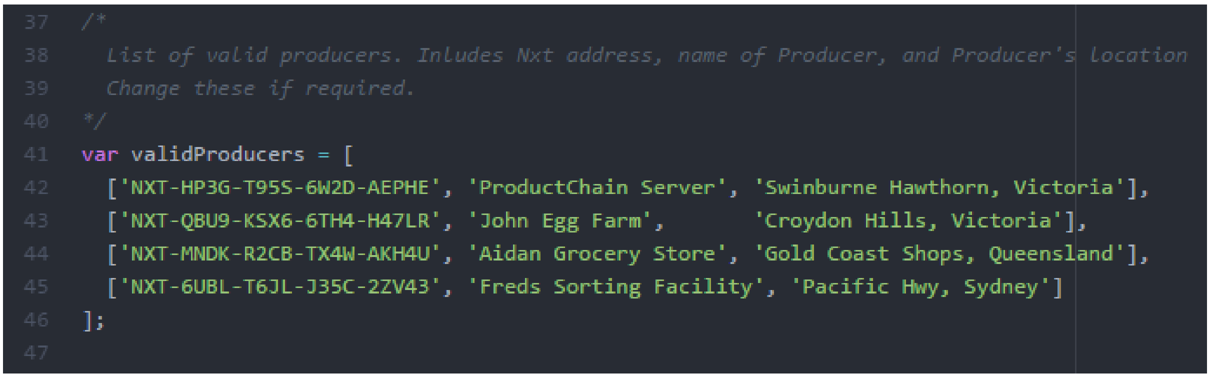


```
/*
  Nxt secret phrase and address of main ProductChain server.
  Change these if required.
*/
var productChainSecretPhrase = "curve excuse kid content gun horse leap poison girlfriend gaze poison comfort";
var productChainAddress = "NXT-HP3G-T95S-6W2D-AEPHE";
```

Finally, you will need to put in the address of the producers you generated in the "Set up Nxt Nodes" documentation on line 41. You will also need to include the ProductChain address in here as well. Also put in a name for the producer, as well as a location. This location isn't used as part of the Proof-of-Location, but is instead used as part of the identification of the producer.

The producers are entered into this array in the following format:

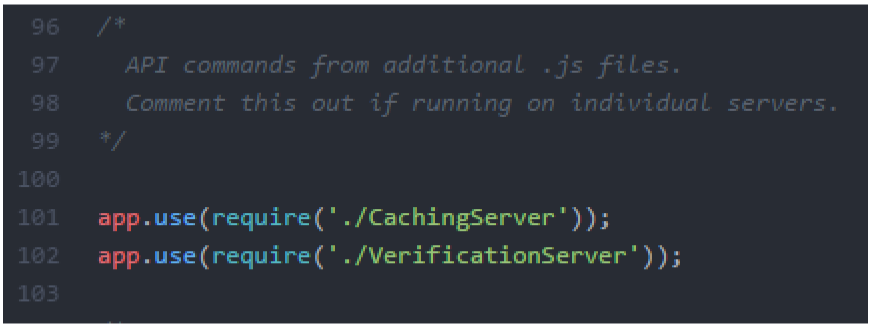
```
['Nxt Address', 'Producer Name', 'Producer Location']
```



```
37  /*
38    List of valid producers. Inludes Nxt address, name of Producer, and Producer's location
39    Change these if required.
40  */
41  var validProducers = [
42    ['NXT-HP3G-T95S-6W2D-AEPHE', 'ProductChain Server', 'Swinburne Hawthorn, Victoria'],
43    ['NXT-QBU9-KSX6-6TH4-H47LR', 'John Egg Farm',       'Croydon Hills, Victoria'],
44    ['NXT-MNDK-R2CB-TX4W-AKH4U', 'Aidan Grocery Store', 'Gold Coast Shops, Queensland'],
45    ['NXT-6UBL-T6JL-J35C-2ZV43', 'Freds Sorting Facility', 'Pacific Hwy, Sydney']
46  ];
47
```

Running all Node.js servers on the same computer

If you wish to run all Node.js servers on the one computer, then you will need to uncomment lines 101 and 102. These are the lines that are used to tell the QR Code Server to include the Caching and Verification servers when it is run.



```
 96  /*
 97    API commands from additional .js files.
 98    Comment this out if running on individual servers.
 99  */
100
101  app.use(require('./CachingServer'));
102  app.use(require('./VerificationServer'));
103
```

Once all additional Node.js servers have been set up, the program can be run by running `node QRCodeServer.js` from the terminal.

Appendix 1 – Trouble with MongoDB

This may resolve issues with starting MongoDB

Open a terminal and enter the following command:

```
sudo nano /etc/systemd/system/mongodb.service
```

Now copy the following text into the newly created file:

```
[Unit]
Description=High-performance, schema-free document-oriented
database
After=network.target

[Service]
User=mongodb
ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

[Install]
WantedBy=multi-user.target
```

You should now be able to start MongoDB through the following command:

```
sudo systemctl start mongodb
```

To permanently enable Mongodb, enter the following command:

```
sudo systemctl enable mongodb
```