Multiple user identity and authentication

Reference links: <https://hyperledger.github.io/composer/tutorials/developer-guide.html>

<https://hyperledger.github.io/composer/integrating/enabling-multiuser.html>

<https://hyperledger.github.io/composer/integrating/enabling-rest-authentication.html>

**Step 0:**

**Install required components:**

npm install -g composer-cli

npm install -g generator-hyperledger-composer

npm install -g composer-rest-server

**Go to fabric-tools folder and in terminal type**

./stopFabric.sh

./teardownFabric.sh

./startFabric.sh

./createFabric.sh

**Get a sample example from git**

git clone <https://github.com/hyperledger/composer-sample-networks.git>

cp -r ./composer-sample-networks/packages/basic-sample-network/ ./my-network

**Update your package.json file in the project folder my-network**

"name": "my-network",

"version": "0.1.6",

"description": "My Commodity Trading network",

"networkImage": "https://hyperledger.github.io/composer-sample-networks/packages/basic-sample-network/networkimage.svg",

"networkImageanimated": "https://hyperledger.github.io/composer-sample-networks/packages/basic-sample-network/networkimageanimated.svg",

"scripts": {

"prepublish": "mkdirp ./dist ; composer archive create --sourceType dir --sourceName . -a ./dist/my-network.bna",

"pretest": "npm run lint",

"lint": "eslint .",

"postlint": "npm run licchk",

"licchk": "license-check",

"postlicchk": "npm run doc",

"doc": "jsdoc --pedantic --recurse -c jsdoc.json",

"test-inner": "mocha -t 0 --recursive && cucumber-js",

"test-cover": "nyc npm run test-inner",

"test": "npm run test-inner"

},

## **Define your Domain (**models/sample.cto**)**

/\*\*

\* My commodity trading network

\*/

namespace org.acme.mynetwork

asset Commodity identified by tradingSymbol {

o String tradingSymbol

o String description

o String mainExchange

o Double quantity

--> Trader owner

}

participant Trader identified by tradeId {

o String tradeId

o String firstName

o String lastName

}

transaction Trade {

--> Commodity commodity

--> Trader newOwner

}

**Write Transaction Processor Functions(**lib/sample.js**)**

/\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

/\*\*

\* Track the trade of a commodity from one trader to another

\* @param {org.acme.mynetwork.Trade} trade - the trade to be processed

\* @transaction

\*/

function tradeCommodity(trade) {

trade.commodity.owner = trade.newOwner;

return getAssetRegistry('org.acme.mynetwork.Commodity')

.then(function (assetRegistry) {

return assetRegistry.update(trade.commodity);

});

}

**Update your Access Control Rules(**permissions.acl**)**

/\*\*

\* Access control rules for mynetwork

\*/

rule Default {

description: "Allow all participants access to all resources"

participant: "ANY"

operation: ALL

resource: "org.acme.mynetwork.\*"

action: ALLOW

}

rule SystemACL {

description: "System ACL to permit all access"

participant: "ANY"

operation: ALL

resource: "org.hyperledger.composer.system.\*\*"

action: ALLOW

}

## **Generate the Business Network Archive**

Switch back to the terminal and type:

cd my-network

npm install

## **Deploy to the running Hyperledger Fabric**

Switch to the terminal, change directory to the dist folder containing the my-network.bna file and type:

cd dist

composer network deploy -a my-network.bna -p hlfv1 -i PeerAdmin -s randomString -A admin -S

**run the admin user using**

**composer-rest-server -p hlfv1 -n my-network -i admin -s adminpw -a true**

**Without adding the users(alice1 and bob1) authenticate the admin with the passport-github strategy(steps include export COMPOSER\_PROVIDERS components as set up in the github)**

**Step 1:**

composer participant add -p hlfv1 -n 'my-network' -i admin -s adminpw -d '{"$class":"org.acme.mynetwork.Trader","tradeId":"kumar1","firstName":" kumar","lastName":"C","company":"Capgemini"}'

composer participant add -p hlfv1 -n 'my-network' -i admin -s adminpw -d '{"$class":"org.acme.mynetwork.Trader","tradeId":"alice1","firstName":" alice","lastName":"vikander","company":"IGate"}'

**Step 2:**

composer identity issue -p hlfv1 -n 'my-network' -i admin -s adminpw -u kumar1 -a "resource:org.acme.mynetwork.Trader#kumar1"

userID = bob1

userSecret = HLEDFPOgPXAk

composer identity issue -p hlfv1 -n 'my-network' -i admin -s adminpw -u alice1 -a "resource:org.acme.mynetwork.Trader#alice1"

userID = alice1

userSecret = DRisDifWFJWx﻿

**Step 3:**

**For Bob**

composer-rest-server -p hlfv1 -n test-network -i bob1 -s CjwexkgepZHm

﻿Github

Client ID

﻿ ﻿﻿d8cc59a17a92649025ee

Client Secret

﻿ ﻿﻿7601b61d4bc97733470db915e1abb01a9da45693

﻿

export COMPOSER\_PROVIDERS='{

"github": {

"provider": "github",

"module": "passport-github",

"clientID": "d8cc59a17a92649025ee",

"clientSecret": "﻿7601b61d4bc97733470db915e1abb01a9da45693",

"authPath": "/auth/github",

"callbackURL": "/auth/github/callback",

"successRedirect": "/",

"failureRedirect": "/"

}

}'

**For alice**

composer-rest-server -p hlfv1 -n test-network -i alice1 -s AfnYpEcqYBHM

Github

Client ID

a8ab616e6dea2e92ea04

Client Secret

62e300b62c5f5005c8fcf103af6eefb57ddff48b

export COMPOSER\_PROVIDERS='{

"github": {

"provider": "github",

"module": "passport-github",

"clientID": "﻿b569b96eaba083a781af",

"clientSecret": "﻿63910d06ecc91bb738dca7f7237eb3c4e652edf9",

"authPath": "/auth/github",

"callbackURL": "/auth/github/callback",

"successRedirect": "/",

"failureRedirect": "/"

}

}'

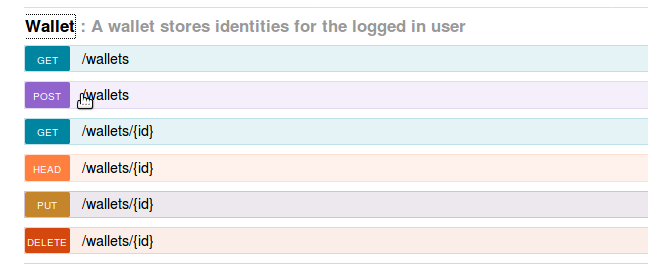
**Step 4:**

Now run the rest server in multiuser mode

﻿

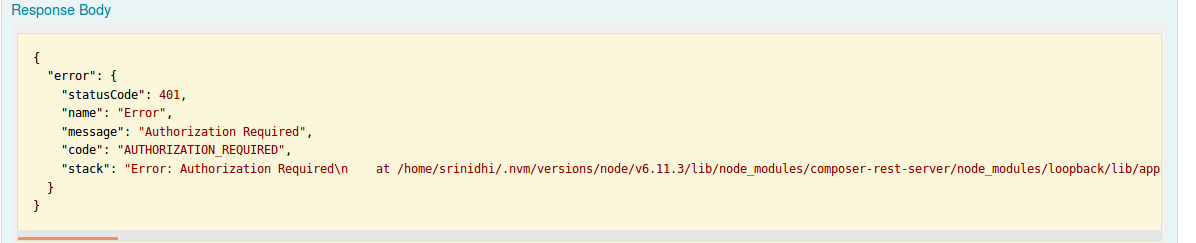
composer-rest-server -p hlfv1 -n test-network -i admin -s adminpw -a true –m true

It will loopback and add a wallet endpoint to the rest collections



***Now getting the /wallets operation will be rejected and asks for authorization (which indicates successful authentication has been done )***

﻿

**

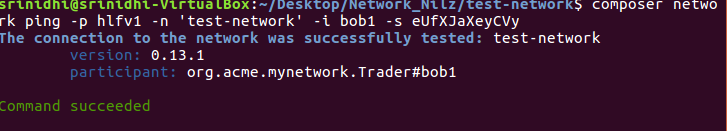
**N.B. [**

Check if the identity is added and issued to blockchain or not

*composer network ping -p hlfv1 -n 'test-network' -i admin -s adminpw*

*composer network ping -p hlfv1 -n 'test-network' -i bob1 -s* iVxxvgslkiIw

*composer network ping -p hlfv1 -n 'test-network' -i alice1 -s* AfnYpEcqYBHM



If not add the user using the ***step 2*** above.

**]**

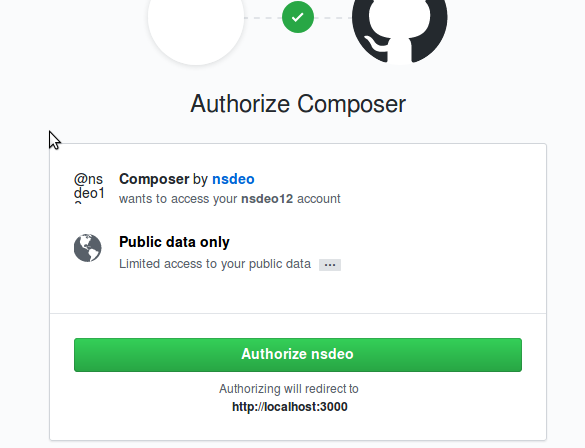
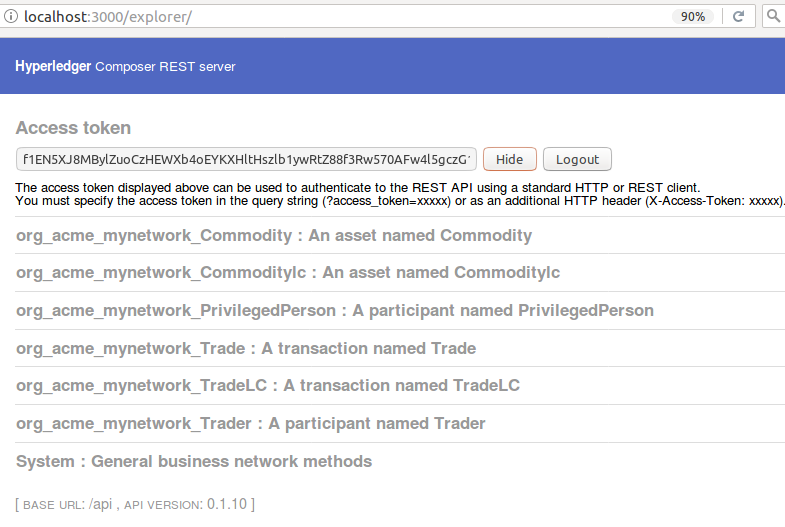
**Step 5:**

**Now login as bob1 as below to authorize the application**

composer-rest-server -p hlfv1 -n test-network -i bob1 -s iVxxvgslkiIw -a true -m true

**Now go to the link below**

**It will show the github screen asking for authorization.so authorize it.**

****

**Step 6:**

**Now login as alice1 and authorize from another github account**

composer-rest-server -p hlfv1 -n test-network -i alice1 -s AfnYpEcqYBHM -a true -m true

**Now go to the link below**

[**http://localhost:3000/auth/github**](http://localhost:3000/auth/github)

**It will show the github screen asking for authorization.so authorize it.**



Now go to the wallet and check for the stored credential using ﻿

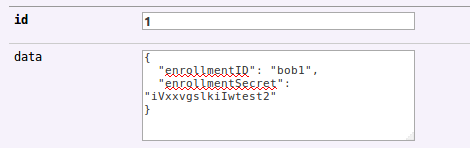
http://localhost:3000/api/wallets

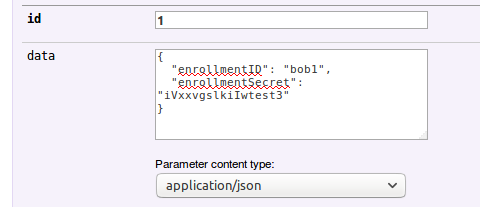


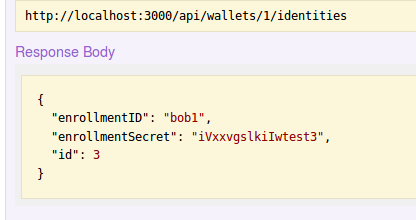
A new identity can be added using the following url too.

****

Can’t follow the below post







﻿Authorizing admin with gihub

Github credential

**Client ID**

ec06841af2547dccc84a

**Client Secret**

2b136dd4338e111623a31f62ebc522778eaf373d

﻿

export COMPOSER\_PROVIDERS='{

"github": {

"provider": "github",

"module": "passport-github",

"clientID": "ec06841af2547dccc84a",

"clientSecret": "2b136dd4338e111623a31f62ebc522778eaf373d",

"authPath": "/auth/github",

"callbackURL": "/auth/github/callback",

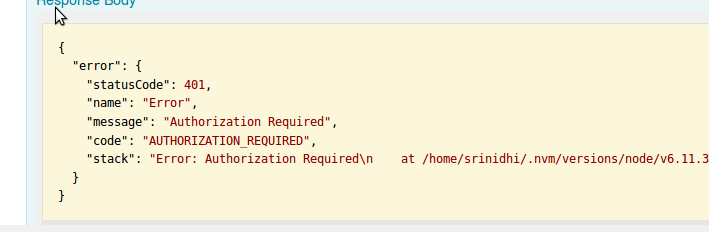
"successRedirect": "/",

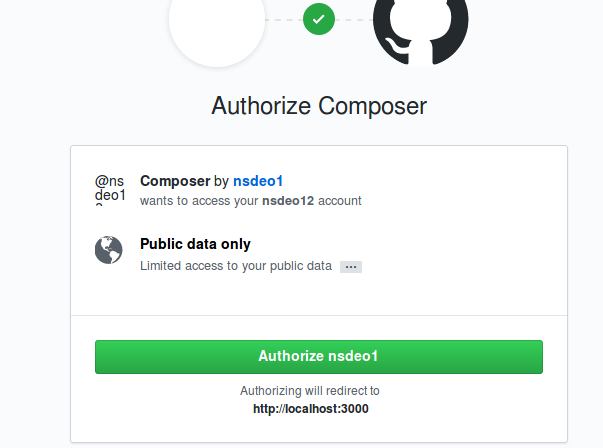
"failureRedirect": "/"

}

}'

Admin is not authorized if not linked to oauth





curl -v http://localhost:3000/api/system/ping?access\_token= FR2oQjhr6u9VxxtPQW6HVU3qK6lG2PItBGem16pGoqICvgj5tnSFAr3YZ1yrMSkB

﻿

curl -v -H 'X-Access-Token: FR2oQjhr6u9VxxtPQW6HVU3qK6lG2PItBGem16pGoqICvgj5tnSFAr3YZ1yrMSkB' http://localhost:3000/api/system/ping

﻿

composer-rest-server -p hlfv1 -n my-network -i admin -s adminpw -m true

﻿

composer identity list -p hlfv1 -n ‘test-network’ -i admin -s adminpw