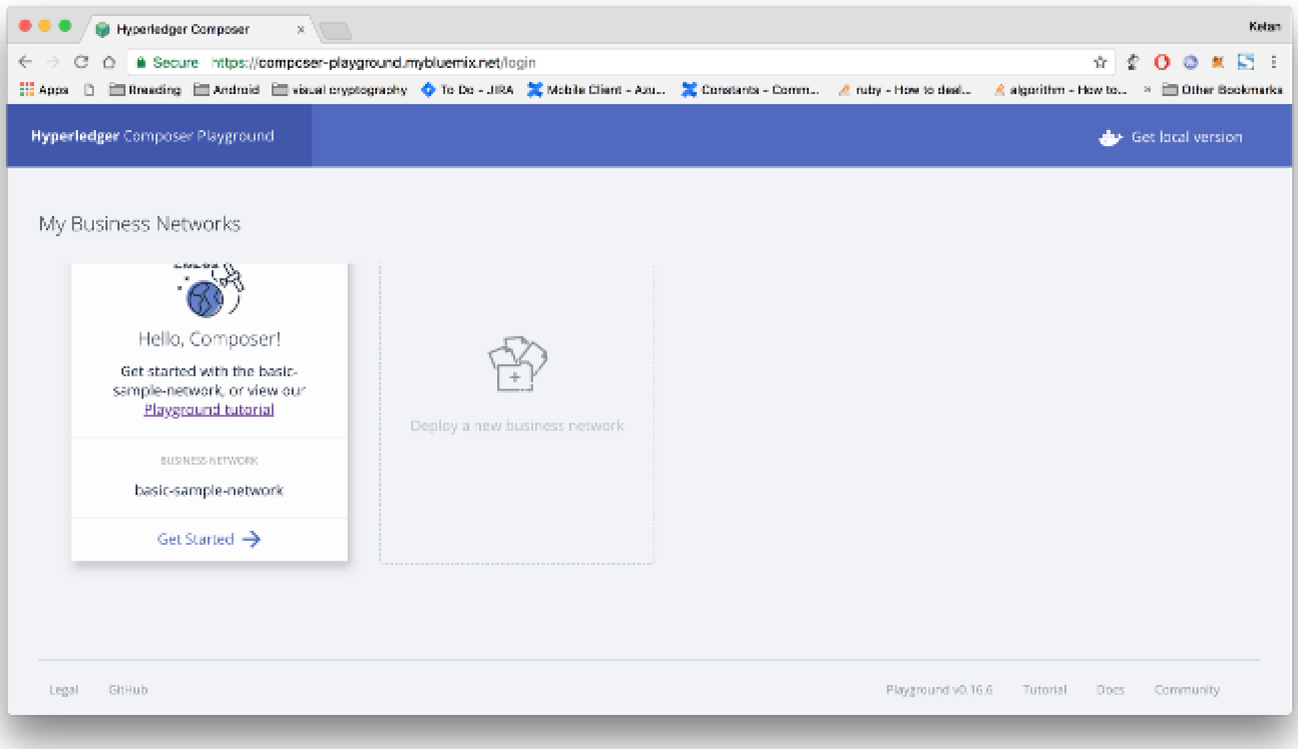
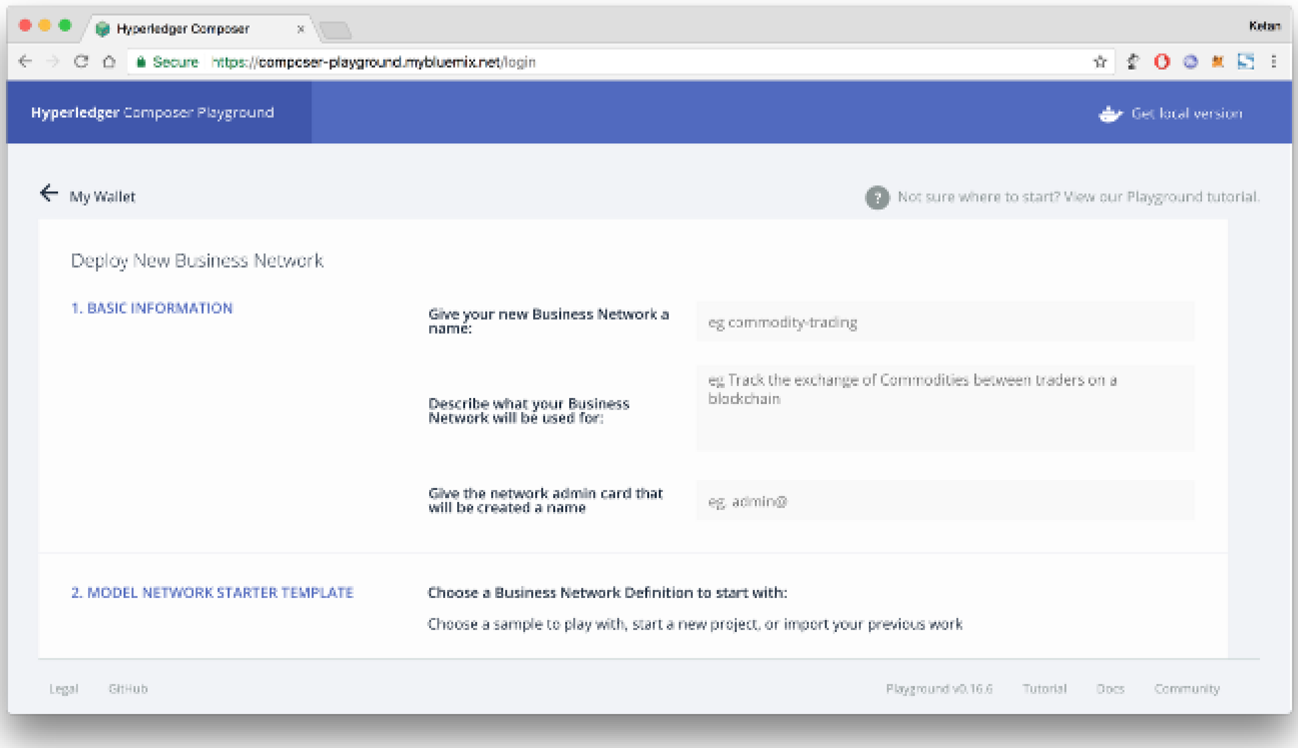
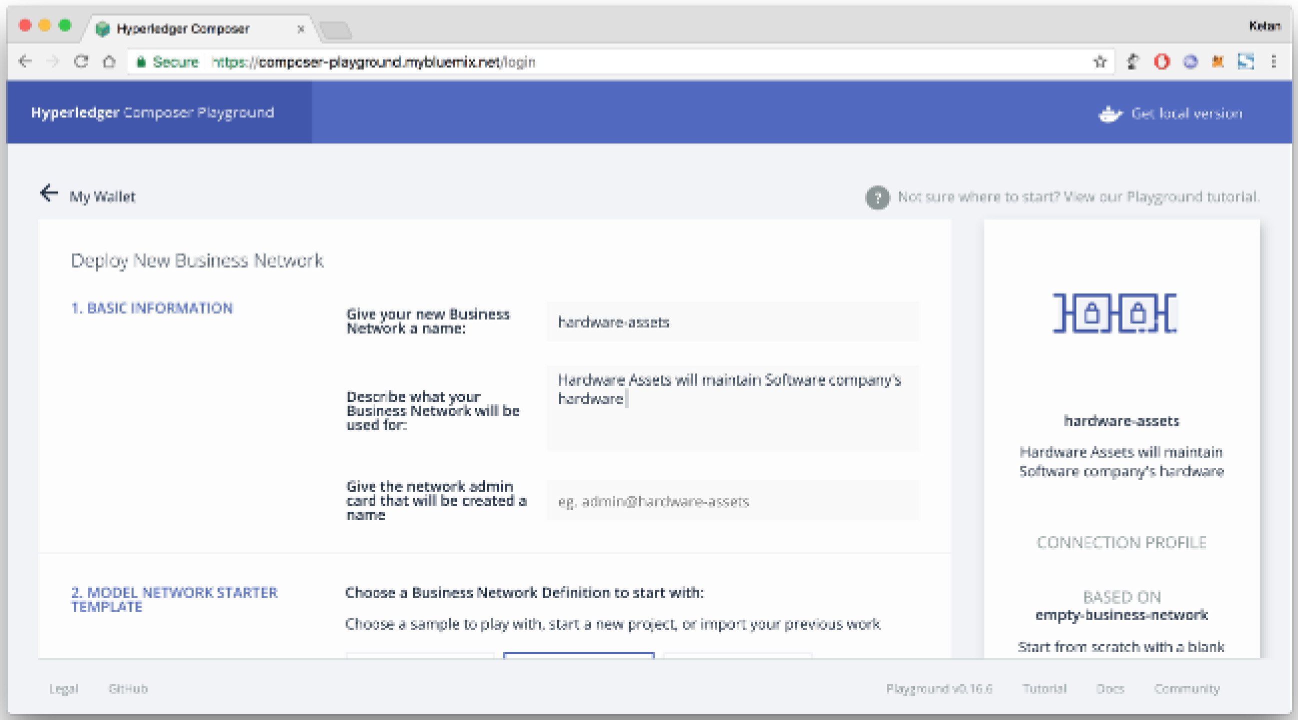

Hyperledger Composer
Secure https://composer-playground.mybluemix.net/login
Hyperledger Composer Playground
Get local version
My Business Networks
Hello, Composer!
Get started with the basic-sample-network, or view our Playground tutorial
BUSINESS NETWORK
basic-sample-network
Get Started
Deploy a new business network
Legal
GitHub
Playground v0.16.6
Tutorial
Docs
Community

Hyperledger Composer Playground Online version

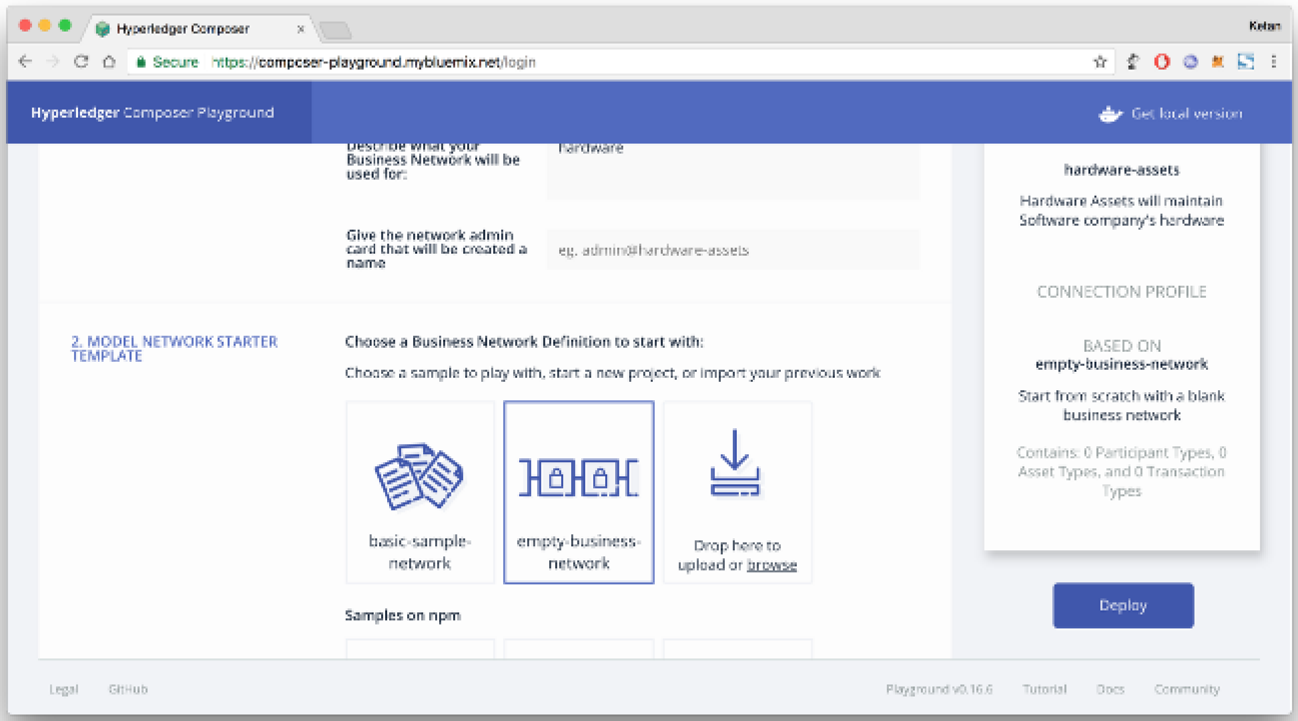
Step 2: Select empty business network


Hyperledger Composer
Secure https://composer-playground.mybluemix.net/login
Hyperledger Composer Playground
Get local version
My Wallet
Not sure where to start? View our Playground tutorial.
Deploy New Business Network
1. BASIC INFORMATION
Give your new Business Network a name:
eg commodity-trading
Describe what your Business Network will be used for:
eg Track the exchange of Commodities between traders on a blockchain
Give the network admin card that will be created a name
eg admin@
2. MODEL NETWORK STARTER TEMPLATE
Choose a Business Network Definition to start with:
Choose a sample to play with, start a new project, or import your previous work
Legal
GitHub
Playground v0.16.6
Tutorial
Docs
Community

Step 3: Fill basic information, select empty business network and click “deploy” button from right pannel

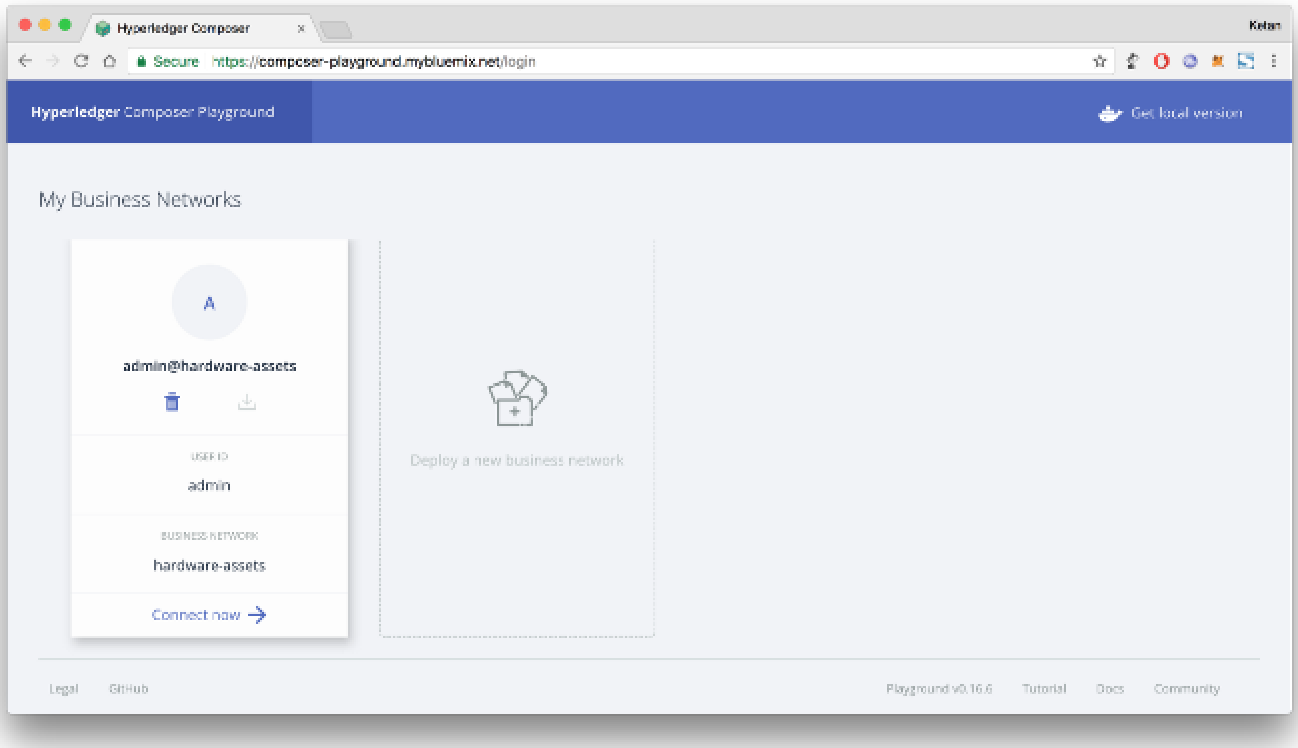

Hyperledger Composer
https://composer-playground.mybluemix.net/login
Hyperledger Composer Playground
Get local version
My Wallet
Not sure where to start? View our Playground tutorial.
Deploy New Business Network
1. BASIC INFORMATION
Give your new Business Network a name: hardware-assets
Describe what your Business Network will be used for: Hardware Assets will maintain Software company's hardware
Give the network admin card that will be created a name: eg. admin@hardware-assets
2. MODEL NETWORK STARTER TEMPLATE
Choose a Business Network Definition to start with:
Choose a sample to play with, start a new project, or import your previous work
hardware-assets
Hardware Assets will maintain Software company's hardware
CONNECTION PROFILE
BASED ON
empty-business-network
Start from scratch with a blank
Legal GitHub Playground v0.16.6 Tutorial Docs Community

Fill basic information

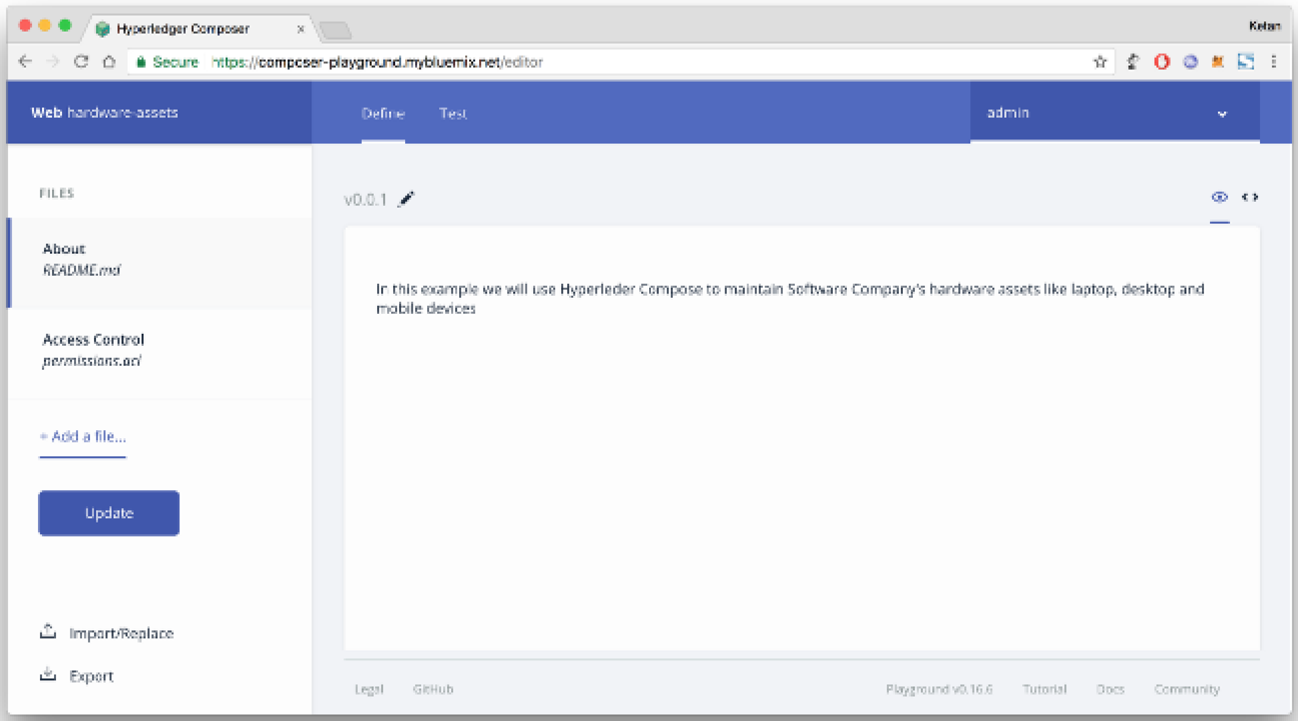

Hyperledger Composer
https://composer-playground.mybluemix.net/login
Hyperledger Composer Playground
Get local version
Describe what your Business Network will be used for: hardware
Give the network admin card that will be created a name: eg. admin@hardware-assets
2. MODEL NETWORK STARTER TEMPLATE
Choose a Business Network Definition to start with:
Choose a sample to play with, start a new project, or import your previous work
basic-sample-network
empty-business-network
Drop here to upload or browse
Samples on npm
hardware-assets
Hardware Assets will maintain Software company's hardware
CONNECTION PROFILE
BASED ON
empty-business-network
Start from scratch with a blank business network
Contains: 0 Participant Types, 0 Asset Types, and 0 Transaction Types
Deploy
Legal GitHub Playground v0.16.6 Tutorial Docs Community

select empty business network

Step 4: Connect to “hardware-assets” business network that we have just deploied


Hyperledger Composer
Secure https://composer-playground.mybluemix.net/login
Hyperledger Composer Playground
Get local version
My Business Networks
admin@hardware-assets
USER ID
admin
BUSINESS NETWORK
hardware-assets
Connect now
Deploy a new business network
Legal GitHub
Playground v0.16.6 Tutorial Docs Community

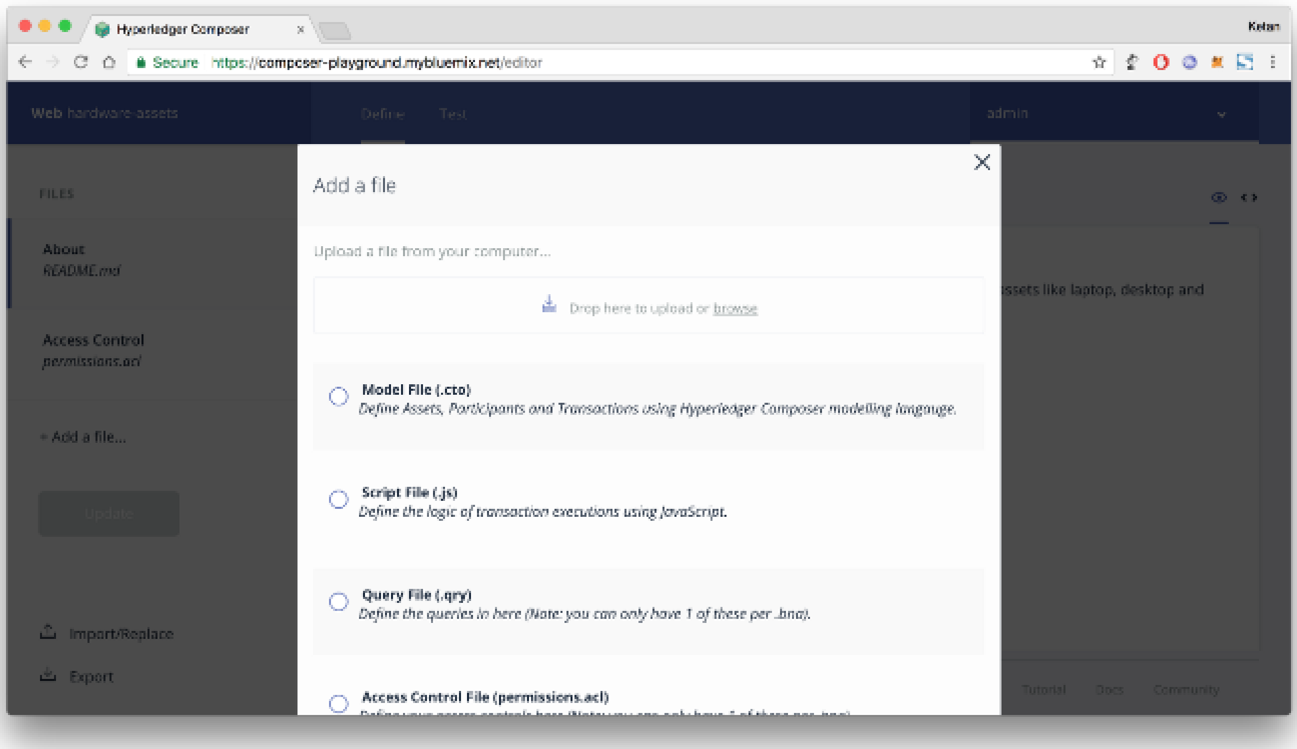
click on "connect now" button



In this example we will use Hyperleder Compose to maintain Software Company's hardware assets like laptop, desktop and mobile devices

Inside hardware-assets business network

Step 5: Click on "+Add a file..." from left panel and select "model file (.cto)"



Add a file

Upload a file from your computer...

Drop here to upload or browse

Model File (.cto)
Define Assets, Participants and Transactions using Hyperledger Composer modelling language.

Script File (.js)
Define the logic of transaction executions using JavaScript.

Query File (.qry)
Define the queries in here (Note: you can only have 1 of these per .bna).

Access Control File (permissions.acl)

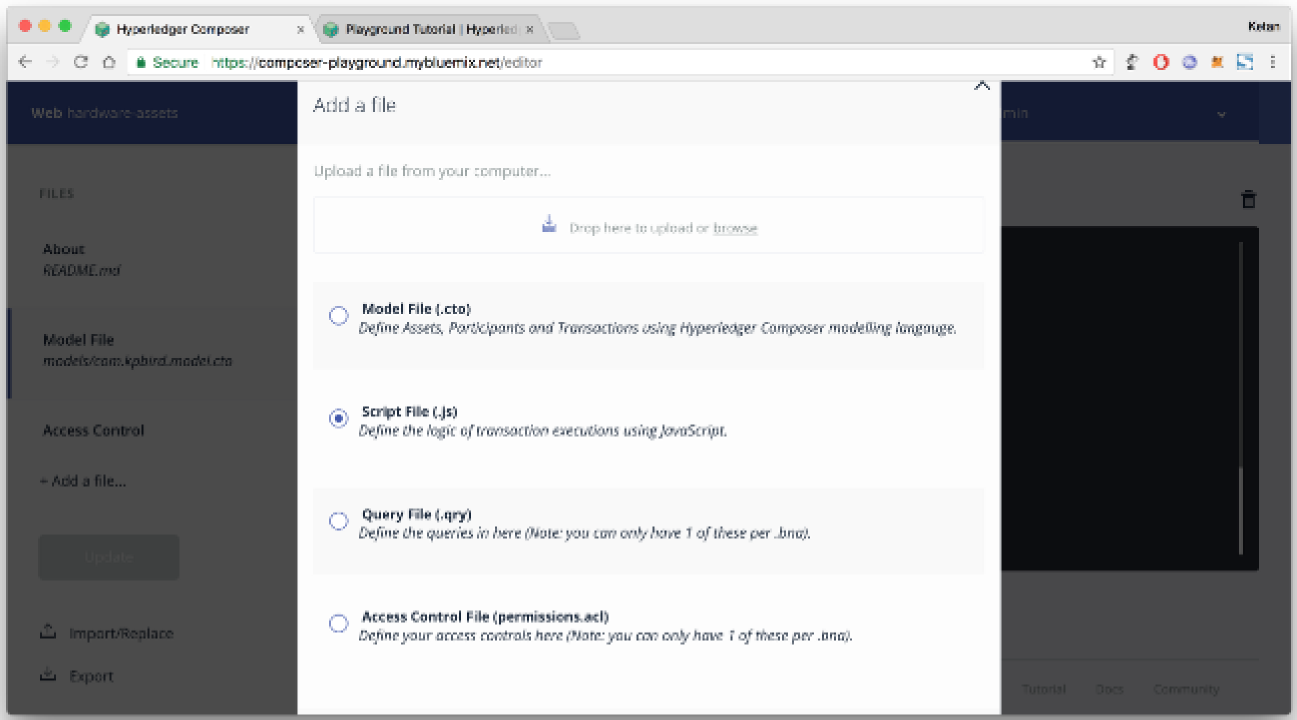
Write following code in model file. Model file contain asset in our case it's hardware, participant in our case participants are employee of organisation and transaction as Allocate hardware to employee. Each model has extra properties. Make sure your have proper and unique namespace. In this example I am using "com.kpbird" as namespace. You can access all models using this namespace i.e. com.kpbird.Hardware, com.kpbird.Employee

```
/**
 * Hardware model
 */namespace com.kpbirdasset Hardware identified by hardwareId {
 o String hardwareId
 o String name
 o String type
 o String description
 o Double quantity
 → Employee owner
}
participant Employee identified by employeeId {
 o String employeeId
 o String firstName
 o String lastName
}
transaction Allocate {
 → Hardware hardware
 → Employee newOwner
}
```

Hyperledger modeling language

reference: https://hyperledger.github.io/composer/reference/cto_language.html

Step 6: Click on "+Add a file..." from left panel and select "script file (*.js)"



Add a file

Upload a file from your computer...

Drop here to upload or browse

Model File (.cto)
Define Assets, Participants and Transactions using Hyperledger Composer modelling language.

[✓ Script File (.js)]
Define the logic of transaction executions using JavaScript.

Query File (.qry)
Define the queries in here (Note: you can only have 1 of these per .bna).

Access Control File (permissions.acl)
Define your access controls here (Note: you can only have 1 of these per .bna).

Write following code in Script File. In Script we can define transaction processing logic. In our case we want to allocate hardware to the employee so, we will update owner of hardware. Make sure about annotation above functions @params and @transaction

```
/**
 * Track the trade of a commodity from one trader to another
 * @param {com.kpbird.Allocate} trade — the trade to be processed
 * @transaction
 */
function allocateHardware(allocate) {
 allocate.hardware.owner = allocate.newOwner;
 return getAssetRegistry('com.kpbird.Hardware')
 .then(function (assetRegistry) {
 return assetRegistry.update(allocate.hardware);
 });
}
```

Hyperledger Composer Script file reference: https://hyperledger.github.io/composer/reference/js_scripts.html

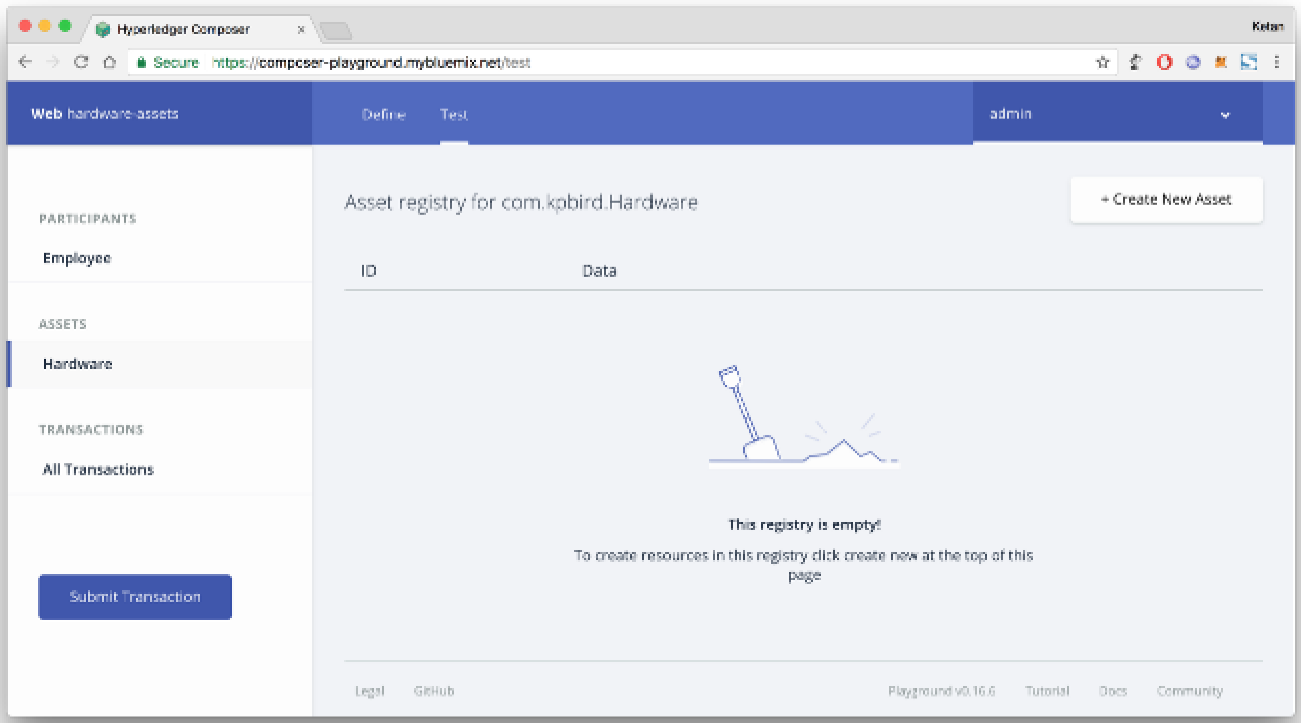
Step 7: permissions.acl file sample is already available, Add following code in permissions.acl file.

```
/**
 * New access control file
 */
 rule AllAccess {
 description: "AllAccess — grant everything to everybody."
 participant: "ANY"
 operation: ALL
 resource: "com.kpbird.**"
 action: ALLOW
 }rule SystemACL{
 description: "System ACL to permit all access"
 participant: "org.hyperledger.composer.system.Participant"
 operation: ALL
 resource: "org.hyperledger.composer.system.**"
 action: ALLOW
}
```

Hyperledger Composer Access Control Language

reference: https://hyperledger.github.io/composer/reference/acl_language.html

Step 8: Now, It's time to test our hardware assets business network. Hyperledger composer gives "Test" facility from composer panel it self. Click on "Test" tab from top panel



Asset registry for com.kpbird.Hardware

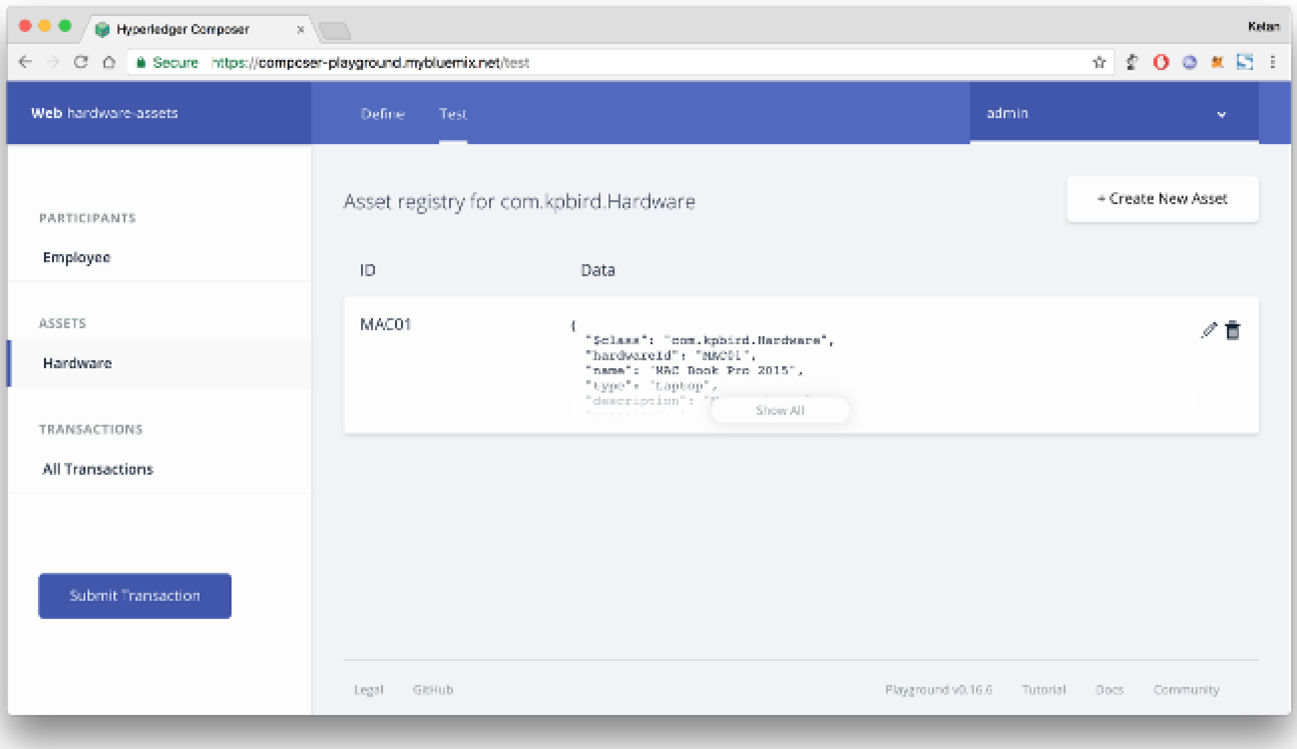
This registry is empty!

To create resources in this registry click create new at the top of this page

Test feature of Hyperledger Composer

Step 9: Create Assets. Click on "Hardware" from left panel and click "+ Create New Assets" from right top corner and add following code. We will create Employee#01 in next step. Click on "Create New" button

```
{
 "$class": "com.kpbird.Hardware",
 "hardwareId": "MAC01",
 "name": "MAC Book Pro 2015",
 "type": "Laptop",
 "description": "Mac Book Pro",
 "quantity": 1,
 "owner": "resource:com.kpbird.Employee#01"
}
```


Asset registry for com.kpbird.Hardware

ID	Data
MAC01	`{ "$class": "com.kpbird.Hardware", "hardwareId": "MAC01", "name": "MAC Book Pro 2015", "type": "Laptop", "description": "`

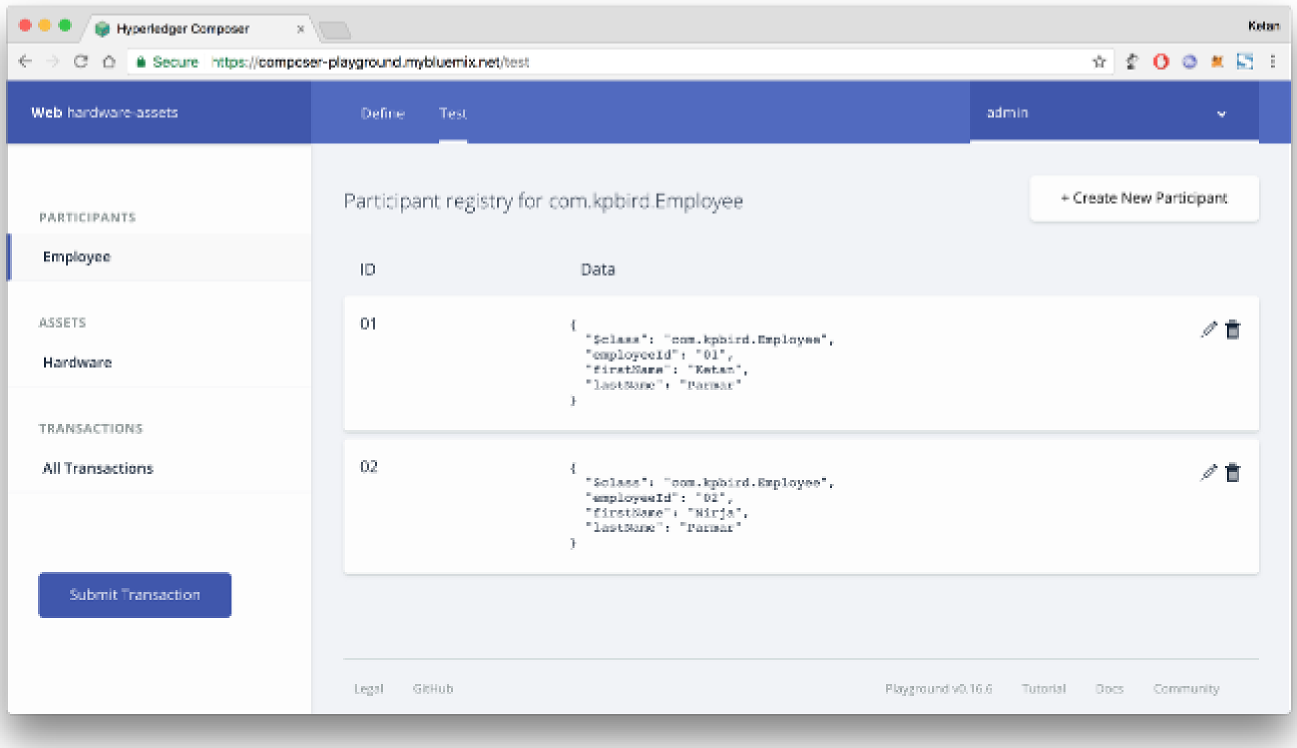
After adding Hardware assets

Steps 10: Let’s create participants. Click “Employee” and click “+ Create New Participants” and add following code. We will add two employees

```
{
 “$class”: “com.kpbird.Employee”,
 “employeeId”: “01”,
 “firstName”: “Ketan”,
 “lastName”: “Parmar”
}
```

Click on “Create New” on dialog

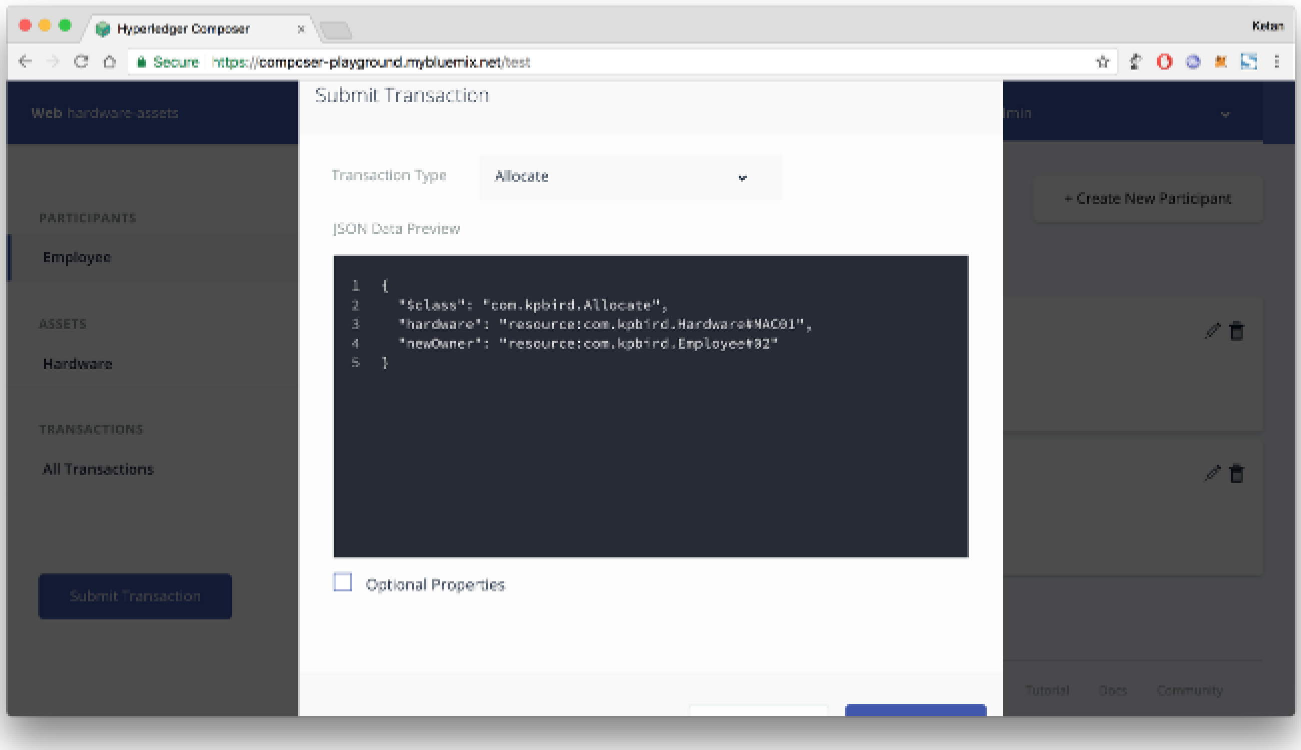
```
{
 “$class”: “com.kpbird.Employee”,
 “employeeId”: “02”,
 “firstName”: “Nirja”,
 “lastName”: “Parmar”
}
```


Participant registry for com.kpbird.Employee

ID	Data
01	`{ "$class": "com.kpbird.Employee", "employeeId": "01", "firstName": "Ketan", "lastName": "Parmar" }`
02	`{ "$class": "com.kpbird.Employee", "employeeId": "02", "firstName": "Nirja", "lastName": "Parmar" }`

We have two employees

Step 11: It’s time to do transaction, We will allocate Macbook Pro from Ketan (Employee#01) to Nirja (Employee#02). Click on “Submit Transaction” button from left panel. In Transaction dialog, We can see all transaction functions on top “Transaction Type” dropdown.

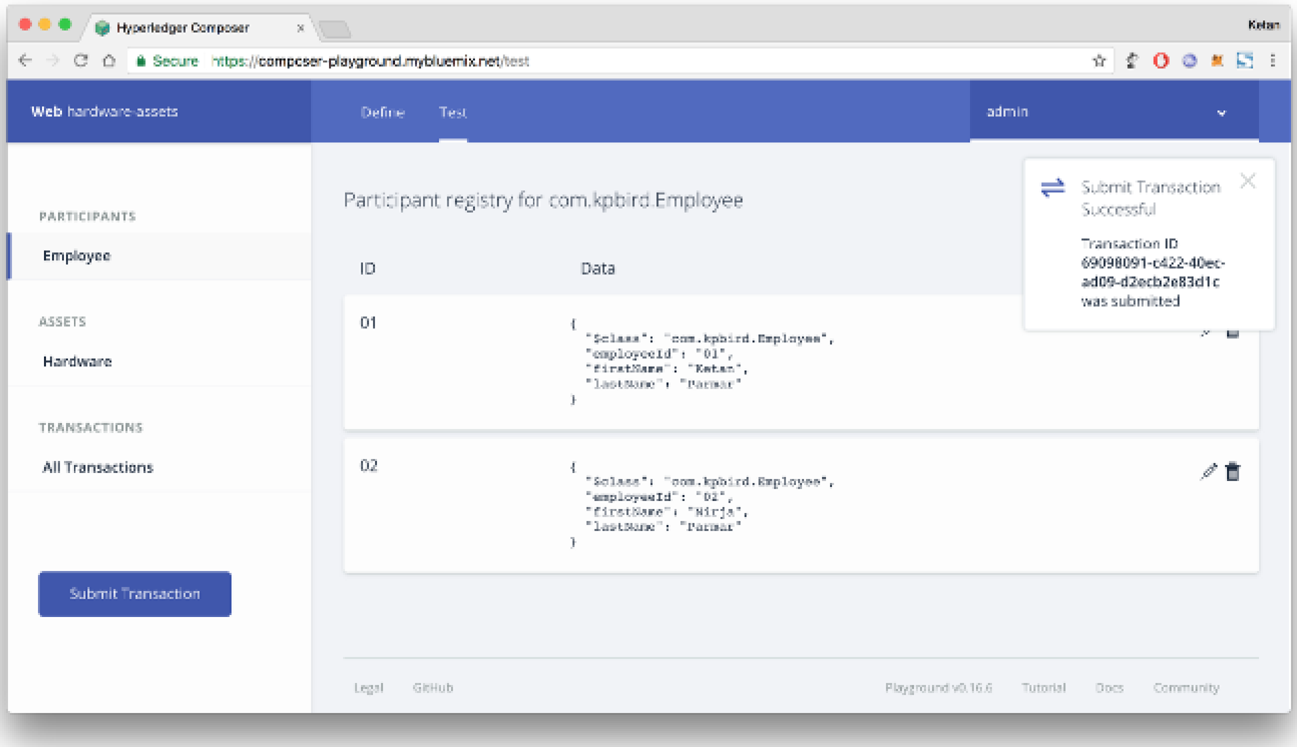


```
{
  "$class": "com.kpbird.Allocate",
  "hardware": "resource:com.kpbird.Hardware#MAC01",
  "newOwner": "resource:com.kpbird.Employee#02"
}
```

Submit Transaction Dialog

```
{
 “$class”: “com.kpbird.Allocate”,
 “hardware”: “resource:com.kpbird.Hardware#MAC01”,
 “newOwner”: “resource:com.kpbird.Employee#02”
}
```

Now, We are allocating Mac01 to Employee 02. Click Submit button after update above JSON in Transaction Dialog. As soon as you hit submit button. Transaction processed and Transaction Id will generate.



Participant registry for com.kpbird.Employee

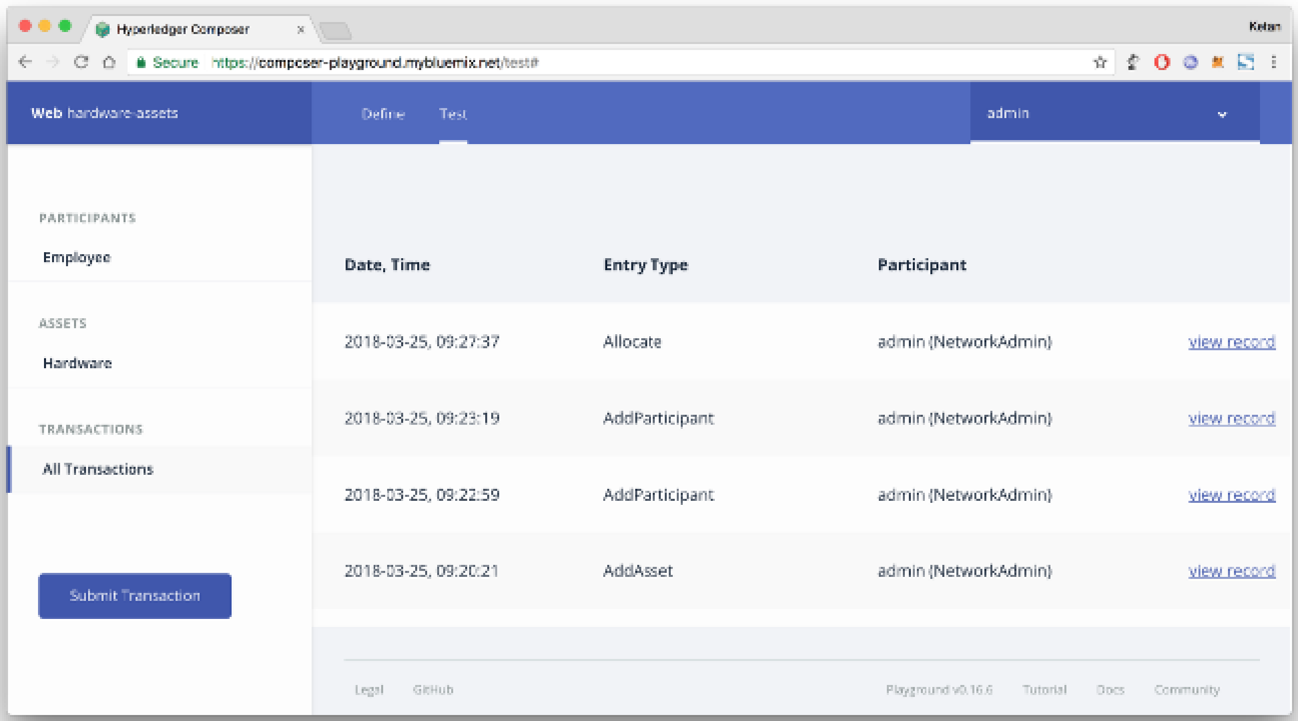
Submit Transaction Successful

Transaction ID 69098091-c422-40ec-ad09-d2ecb2e83d1c was submitted

ID	Data
01	{ "$class": "com.kpbird.Employee", "employeeId": "01", "firstName": "Ketan", "lastName": "Parmar" }
02	{ "$class": "com.kpbird.Employee", "employeeId": "02", "firstName": "Nirja", "lastName": "Parmar" }

.

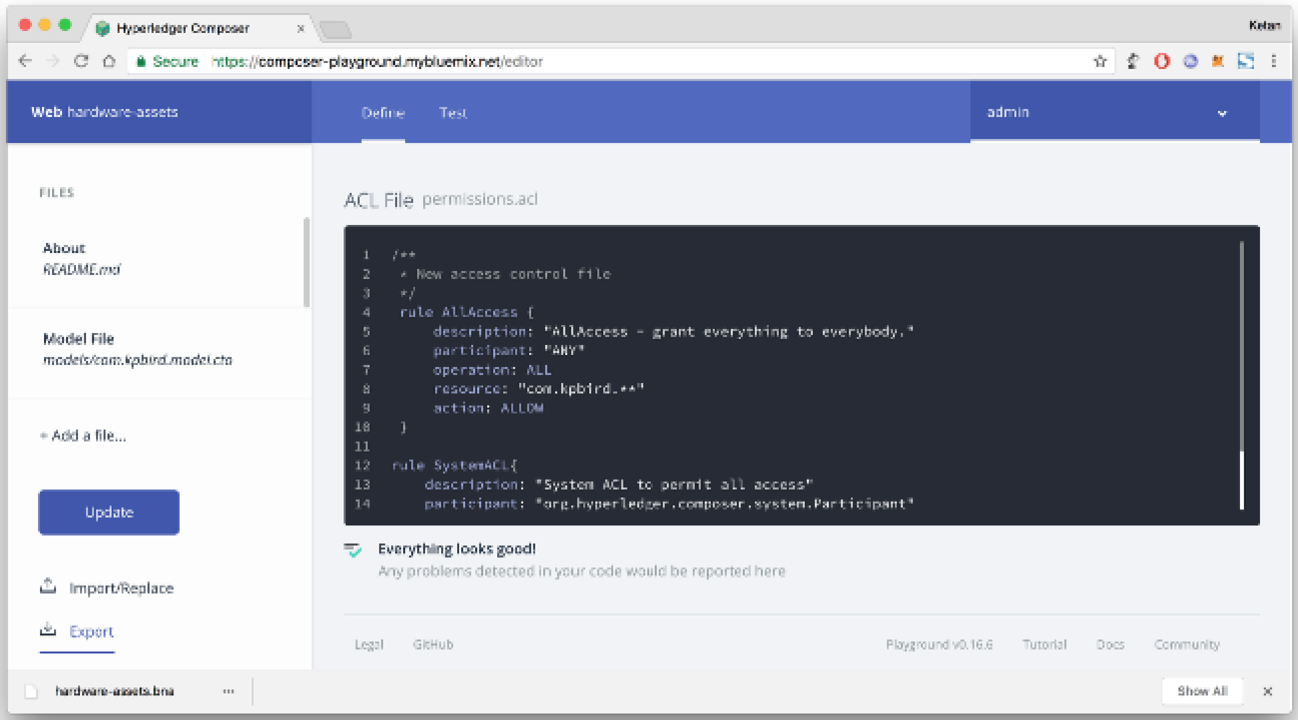
Step 12: Click on “All Transactions” from left panel to verify all transactions. In following screenshots you can see add assets, ass participants and allocation all operation are consider as transactions. “view records” will give us more information about transaction.



Date, Time	Entry Type	Participant	
2018-03-25, 09:27:37	Allocate	admin (NetworkAdmin)	view record
2018-03-25, 09:23:19	AddParticipant	admin (NetworkAdmin)	view record
2018-03-25, 09:22:59	AddParticipant	admin (NetworkAdmin)	view record
2018-03-25, 09:20:21	AddAsset	admin (NetworkAdmin)	view record

All Transactions

Step 13: Now, It’s time to deploy “hardware-assets” business network to Hyperledger Fabric. Click on “Define” tab from top panel and click “Export” button from left panel. Export will create hardware-assets.bna file.



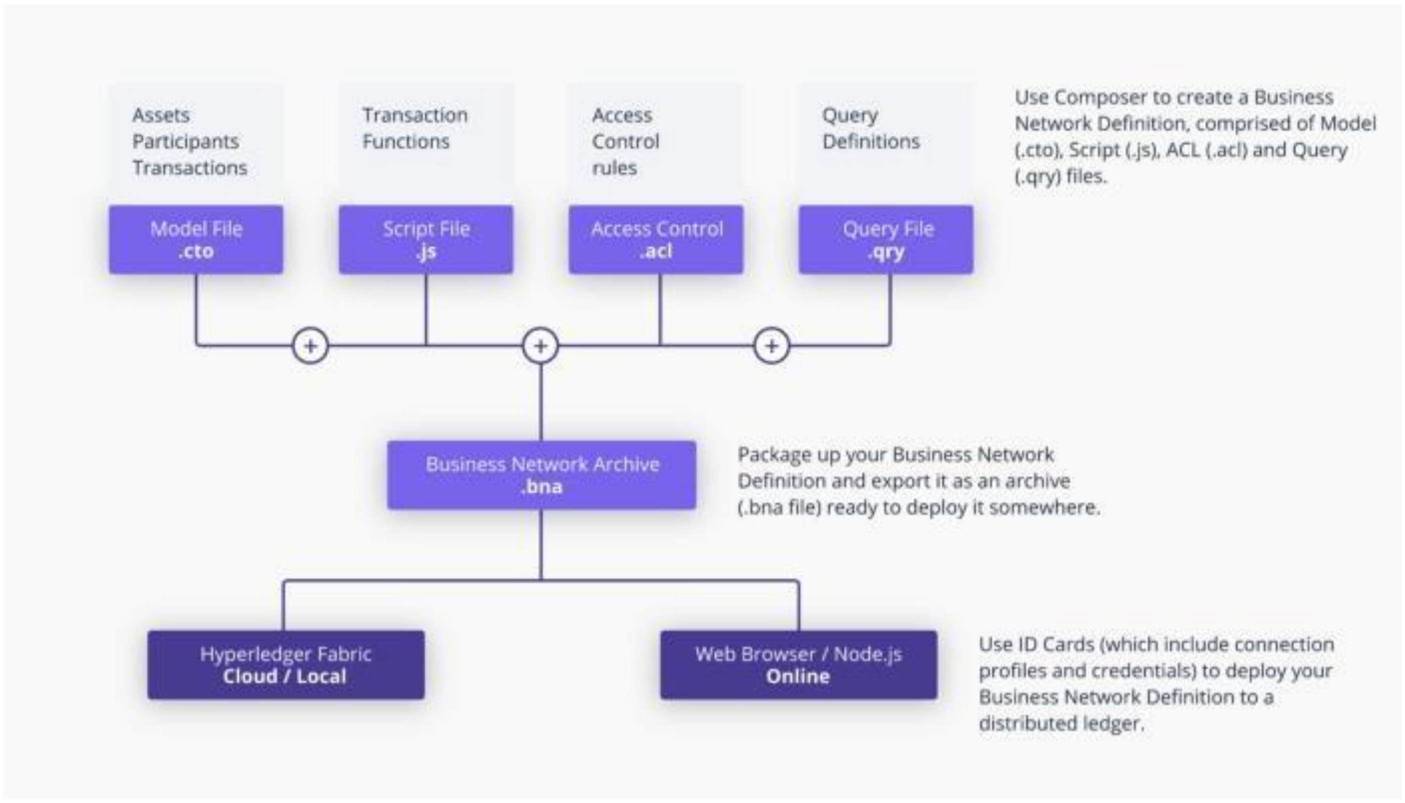
ACL File permissions.acl

```
/**
 * New access control file
 */
 rule AllAccess {
     description: "AllAccess - grant everything to everybody."
     participant: "ANY"
     operation: ALL
     resource: "com.kpbird.**"
     action: ALLOW
 }

rule SystemACL{
    description: "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Participant"
```

Everything looks good!
Any problems detected in your code would be reported here

Download hardware-assets.bna file

.bna is Business Network Archive file which contains model, script, network access and query file

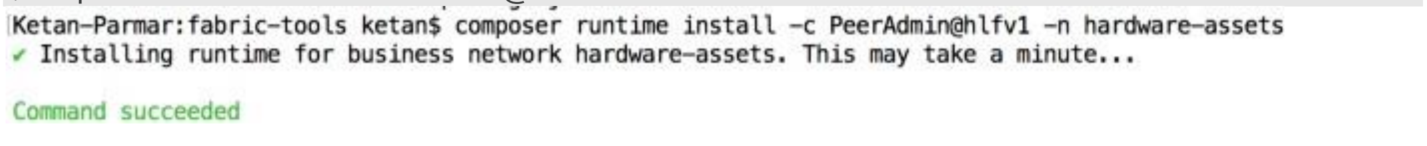

Assets
Participants
Transactions
Transaction
Functions
Access
Control
rules
Query
Definitions
Use Composer to create a Business Network Definition, comprised of Model (.cto), Script (.js), ACL (.acl) and Query (.qry) files.
Model File
.cto
Script File
.js
Access Control
.acl
Query File
.qry
Business Network Archive
.bna
Package up your Business Network Definition and export it as an archive (.bna file) ready to deploy it somewhere.
Hyperledger Fabric
Cloud / Local
Web Browser / Node.js
Online
Use ID Cards (which include connection profiles and credentials) to deploy your Business Network Definition to a distributed ledger.

source: https://hyperledger.github.io/composer/introduction/introduction

Step 14: Start Docker and run following commands from ~/fabric-tools directory

Install business network to Hyperledger Fabric, If business network is already installed you can use "update" instead of "install"

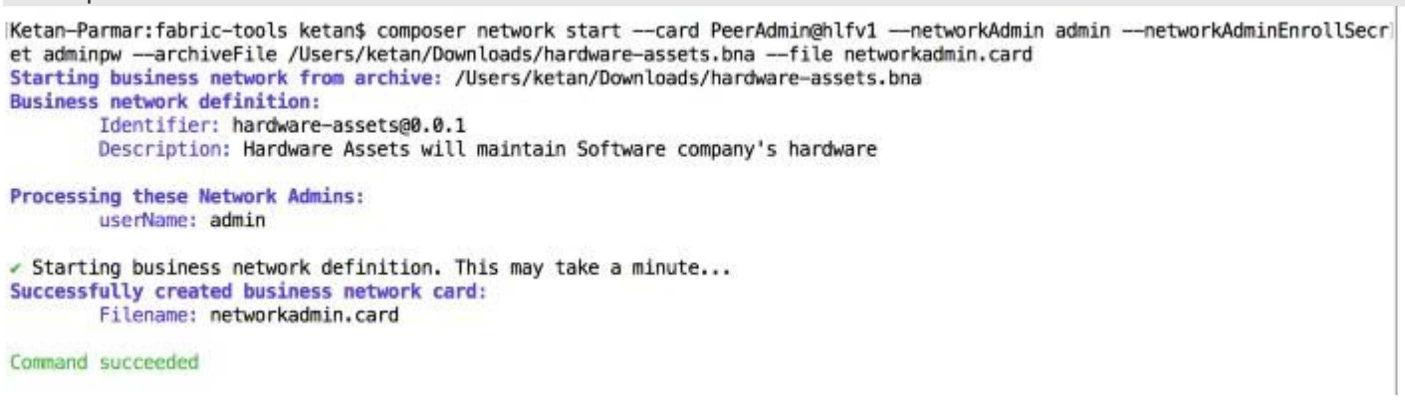
$composer runtime install -c PeerAdmin@hlfv1 -n hardware-assets



```
Ketan-Parmar:fabric-tools ketan$ composer runtime install -c PeerAdmin@hlfv1 -n hardware-assets
✔ Installing runtime for business network hardware-assets. This may take a minute...

Command succeeded
```

Following command will deploy and start hardware-assets.bna file. Change hardware-assets.bna file before you execute following command. networkadmin.card file will generate in ~/fabric-tools directory from previous command.

$composer network start — card PeerAdmin@hlfv1 — networkAdmin admin — networkAdminEnrollSecret adminpw — archiveFile /Users/ketan/Downloads/hardware-assets.bna — file networkadmin.card



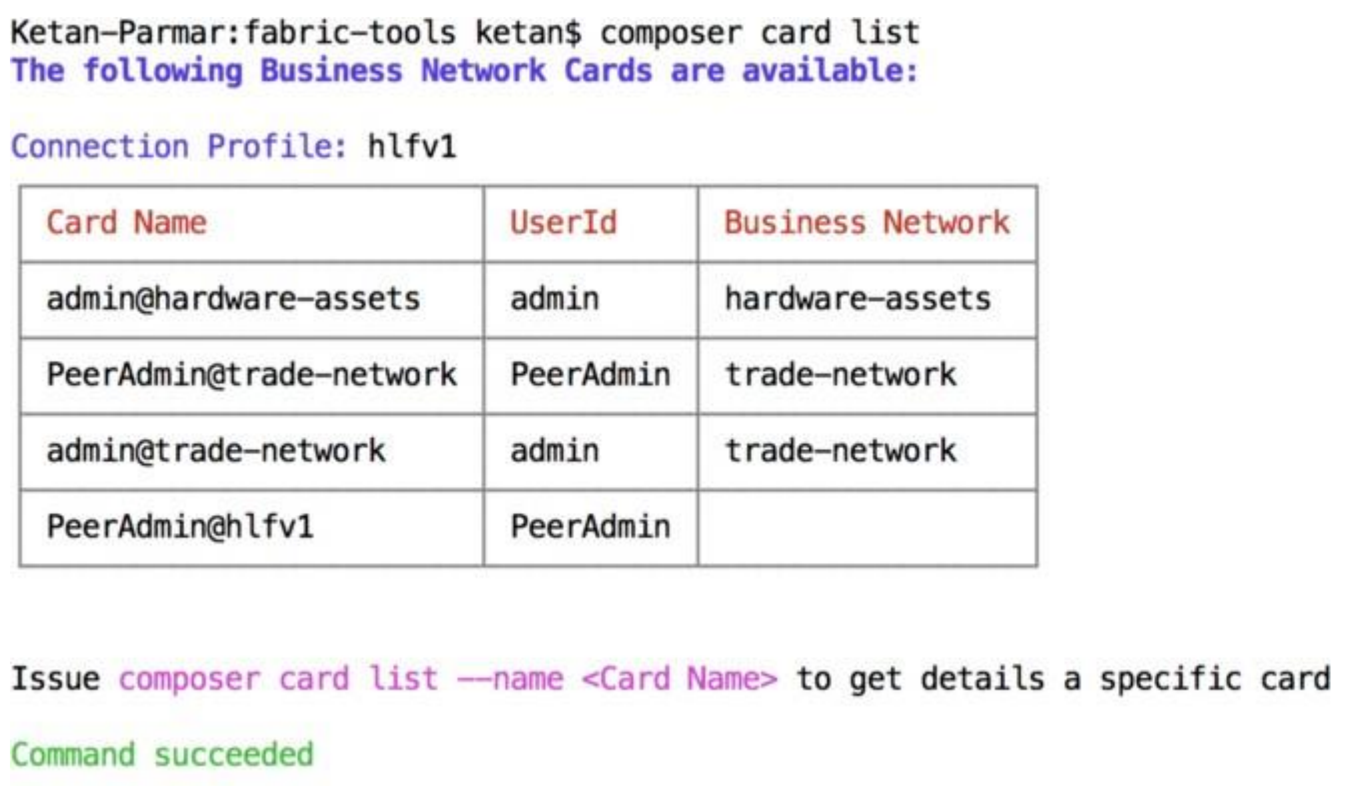
```
Ketan-Parmar:fabric-tools ketan$ composer network start --card PeerAdmin@hlfv1 --networkAdmin admin --networkAdminEnrollSecr
et adminpw --archiveFile /Users/ketan/Downloads/hardware-assets.bna --file networkadmin.card
Starting business network from archive: /Users/ketan/Downloads/hardware-assets.bna
Business network definition:
        Identifier: hardware-assets@0.0.1
        Description: Hardware Assets will maintain Software company's hardware

Processing these Network Admins:
        userName: admin

✔ Starting business network definition. This may take a minute...
Successfully created business network card:
        Filename: networkadmin.card

Command succeeded
```

To connect business network you need connection card. so we can import networkadmin.card using following command

$composer card import -f networkadmin.card

To make sure networkadmin.card successfully install you can list cards using following command

$composer card list

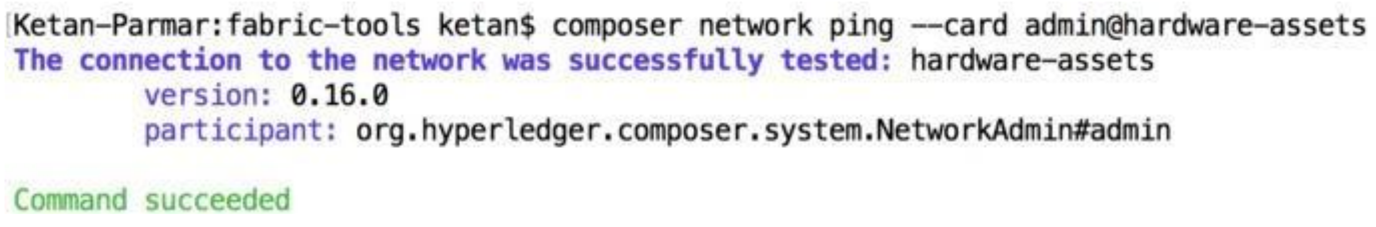


```
Ketan-Parmar:fabric-tools ketan$ composer card list
The following Business Network Cards are available:

Connection Profile: hlfv1
┌─────────────────────────┬───────────┬──────────────────┐
│ Card Name               │ UserId    │ Business Network │
├─────────────────────────┼───────────┼──────────────────┤
│ admin@hardware-assets   │ admin     │ hardware-assets  │
├─────────────────────────┼───────────┼──────────────────┤
│ PeerAdmin@trade-network │ PeerAdmin │ trade-network    │
├─────────────────────────┼───────────┼──────────────────┤
│ admin@trade-network     │ admin     │ trade-network    │
├─────────────────────────┼───────────┼──────────────────┤
│ PeerAdmin@hlfv1         │ PeerAdmin │                  │
└─────────────────────────┴───────────┴──────────────────┘


Issue composer card list --name <Card Name> to get details a specific card

Command succeeded
```

Following command will make sure that our hardware-assets business network is successfully running in

Hyperledger Fabric.

$composer network ping – card admin@hardware-assets

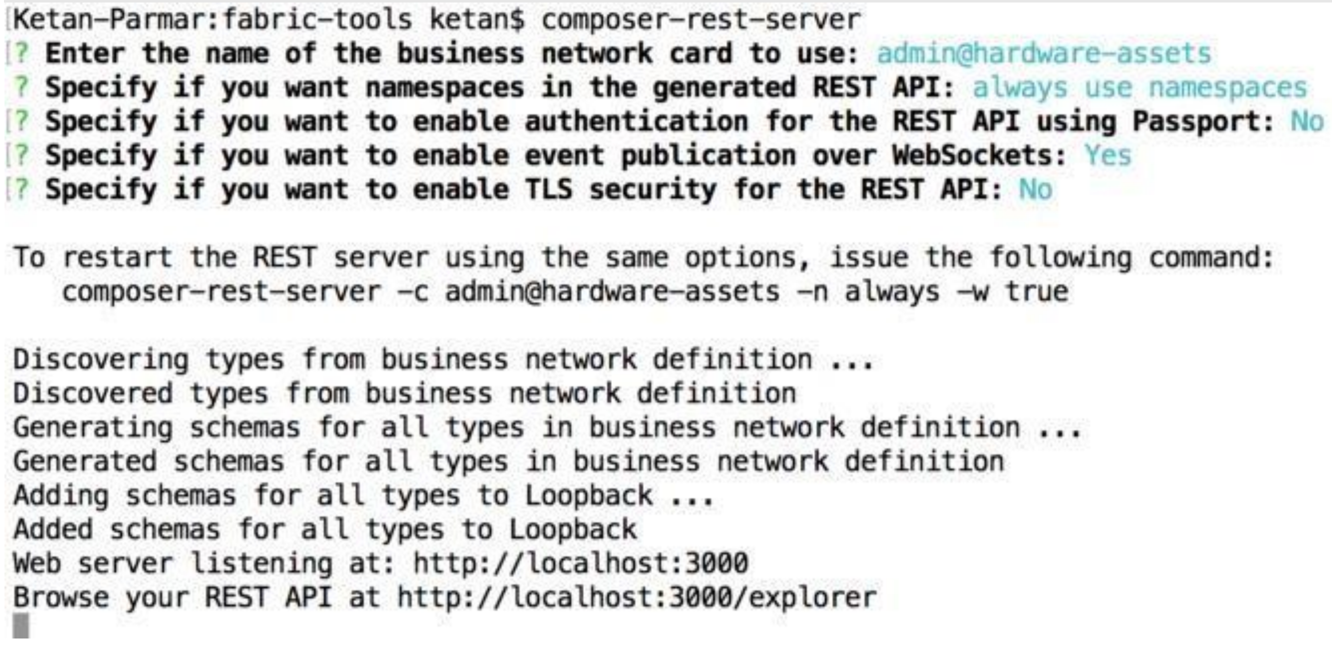


```
Ketan-Parmar:fabric-tools ketan$ composer network ping --card admin@hardware-assets
The connection to the network was successfully tested: hardware-assets
        version: 0.16.0
        participant: org.hyperledger.composer.system.NetworkAdmin#admin

Command succeeded
```

Now It's time to interact with REST API. To develop Web or Mobile Application we require REST API. you car

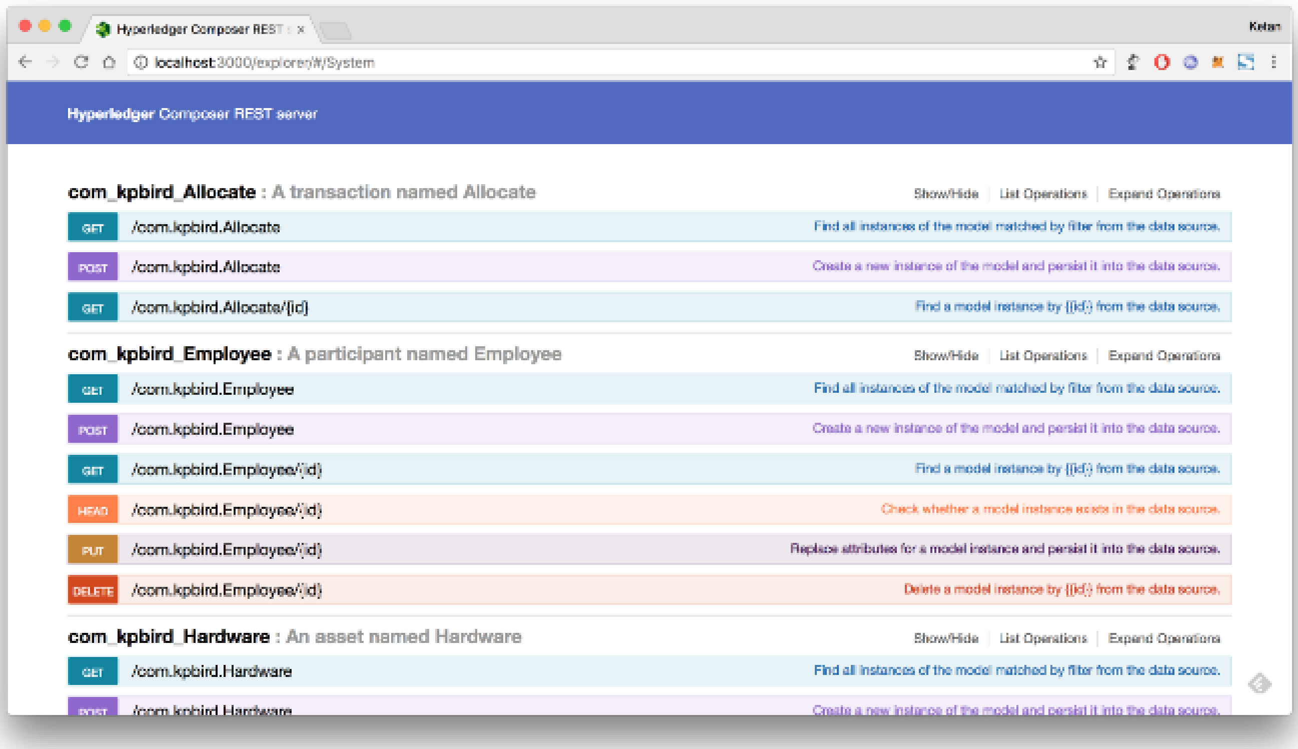
run following command to generate REST API for hardware-assets business network.

```
$composer-rest-server
```



```
Ketan-Parmar:fabric-tools ketan$ composer-rest-server
? Enter the name of the business network card to use: admin@hardware-assets
? Specify if you want namespaces in the generated REST API: always use namespaces
? Specify if you want to enable authentication for the REST API using Passport: No
? Specify if you want to enable event publication over WebSockets: Yes
? Specify if you want to enable TLS security for the REST API: No

To restart the REST server using the same options, issue the following command:
   composer-rest-server -c admin@hardware-assets -n always -w true

Discovering types from business network definition ...
Discovered types from business network definition
Generating schemas for all types in business network definition ...
Generated schemas for all types in business network definition
Adding schemas for all types to Loopback ...
Added schemas for all types to Loopback
Web server listening at: http://localhost:3000
Browse your REST API at http://localhost:3000/explorer
```

rest server will ask few basic information before generate rest api



Hyperledger Composer REST server

com_kpbird_Allocate : A transaction named Allocate Show/Hide | List Operations | Expand Operations

```
GET     /com.kpbird.Allocate              Find all instances of the model matched by filter from the data source.
POST    /com.kpbird.Allocate              Create a new instance of the model and persist it into the data source.
GET     /com.kpbird.Allocate/{id}         Find a model instance by {{id}} from the data source.
```

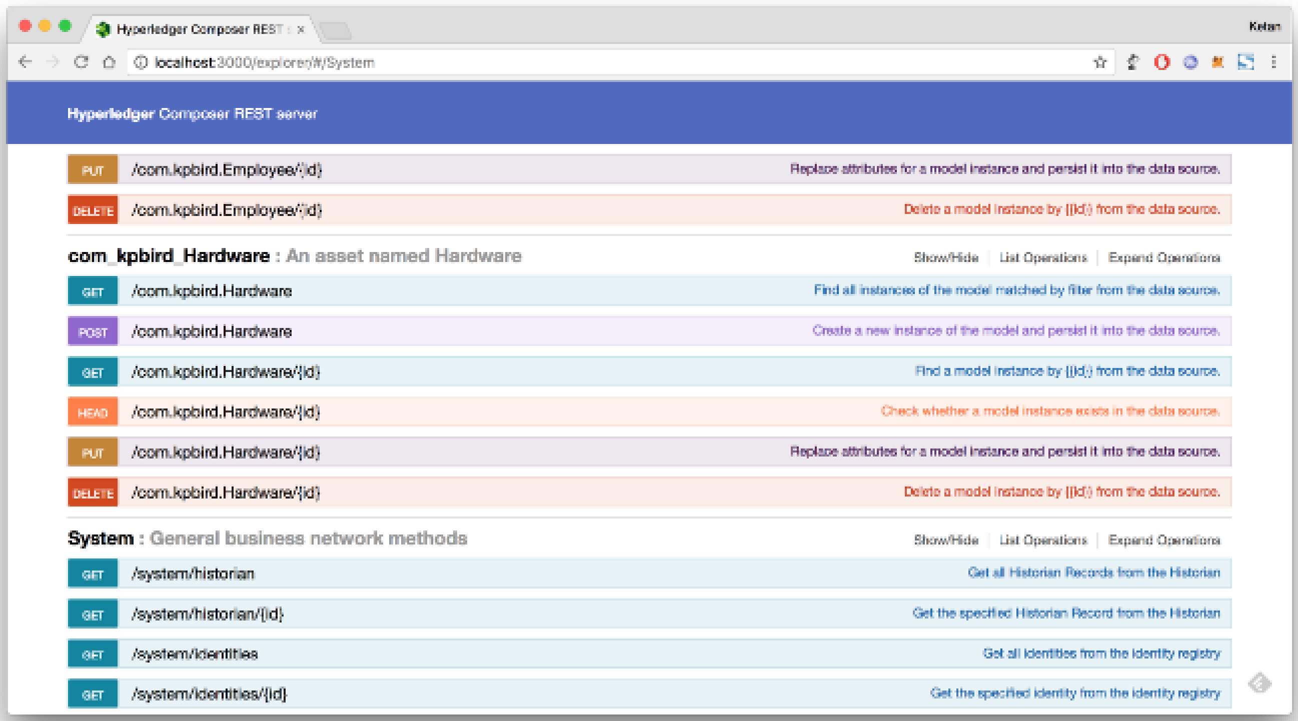
com_kpbird_Employee : A participant named Employee Show/Hide | List Operations | Expand Operations

```
GET     /com.kpbird.Employee              Find all instances of the model matched by filter from the data source.
POST    /com.kpbird.Employee              Create a new instance of the model and persist it into the data source.
GET     /com.kpbird.Employee/{id}         Find a model instance by {{id}} from the data source.
HEAD    /com.kpbird.Employee/{id}         Check whether a model instance exists in the data source.
PUT     /com.kpbird.Employee/{id}         Replace attributes for a model instance and persist it into the data source.
DELETE  /com.kpbird.Employee/{id}         Delete a model instance by {{id}} from the data source.
```

com_kpbird_Hardware : An asset named Hardware Show/Hide | List Operations | Expand Operations

```
GET     /com.kpbird.Hardware              Find all instances of the model matched by filter from the data source.
POST    /com.kpbird.Hardware              Create a new instance of the model and persist it into the data source.
```

REST API for our hardware assets



Hyperledger Composer REST server

```
PUT     /com.kpbird.Employee/{id}         Replace attributes for a model instance and persist it into the data source.
DELETE  /com.kpbird.Employee/{id}         Delete a model instance by {{id}} from the data source.
```

com_kpbird_Hardware : An asset named Hardware Show/Hide | List Operations | Expand Operations

```
GET     /com.kpbird.Hardware              Find all instances of the model matched by filter from the data source.
POST    /com.kpbird.Hardware              Create a new instance of the model and persist it into the data source.
GET     /com.kpbird.Hardware/{id}         Find a model instance by {{id}} from the data source.
HEAD    /com.kpbird.Hardware/{id}         Check whether a model instance exists in the data source.
PUT     /com.kpbird.Hardware/{id}         Replace attributes for a model instance and persist it into the data source.
DELETE  /com.kpbird.Hardware/{id}         Delete a model instance by {{id}} from the data source.
```

System : General business network methods Show/Hide | List Operations | Expand Operations

```
GET     /system/historian                 Get all Historian Records from the Historian
GET     /system/historian/{id}            Get the specified Historian Record from the Historian
GET     /system/identities                Get all identities from the identity registry
GET     /system/identities/{id}           Get the specified identity from the identity registry
```

REST API methods for all operations

Conclusion: In this way we have learnt about hyperledger and its use case in business world.
