Lab 3. Using Hyperledger Composer

In this lab, we will use a GUI front end called *Composer* to build networks on the Fabric blockchain.

First, let's shut down the existing docker images.

```
$ docker ps -aq | xargs docker rm -f
$ docker images -aq | xargs docker rmi -f
```

Now, let's install Composer.

```
curl -sSL https://hyperledger.github.io/composer/install-hlfv1.sh |
bash
```

Open a browser window and navigate to the IP address of your server and port 8080.

This should bring up the hyperledger composer window.

Next, we want to create a new business network from scratch. A business network has a couple of defining properties; a name, and an optional description. You can also choose to base a new business network on an existing template, or import your own template.

Click Deploy a new business network under the Web Browser heading to get started.

The new business network needs a name, let's call it acme_widgets

Optionally, you can enter a description for your business network.

Next we must select a business network to base ours on, because we want to build the network from scratch, click empty-business-network.

Now that our network is defined, click Deploy.

Now that we've created and deployed the business network, you should see a new business network card called admin for our business network tutorial-network in your wallet. The wallet can contain business network cards to connect to multiple deployed business networks.

When connecting to an external blockchain, business network cards represent everything necessary to connect to a business network. They include connection details, authentication material, and metadata.

To connect to our business network click Connect now under our business network card.

As you can see, we're in the Define tab right now, this tab is where you create and edit the files that make up a business network definition, before deploying them and testing them using the Test tab.

As we selected an empty business network template, we need to define our business network files. The first step is to add a model file. Model files define the assets, participants, transactions, and events in our business network.

Click the Add a file button.

Click the Model file and click Add.

Delete the lines of code in the model file and replace it with this:

```
/**
 *Acme Widgets Trading platform
 */
namespace com.acmewidgets.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
}
```

Now that the domain model has been defined, we can define the transaction logic for the business network. Composer expresses the logic for a business network using JavaScript functions. These functions are automatically executed when a transaction is submitted for processing.

Click the Add a file button.

Click the Script file and click Add.

Delete the lines of code in the script file and replace it with the following code:

```
/**
 * Track the trade of a commodity from one trader to another
 * @param {com.acmewidgets.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);
        });
}
```

This function simply changes the owner property on a commodity based on the newOwner property on an incoming Trade transaction. It then persists the modified Commodity back into the asset registry, used to store Commodity instances.

Access control files define the access control rules for business networks. Our network is simple, so the default access control file doesn't need editing. The basic file gives the current participant networkAdmin full access to business network and system-level operations.

While you can have multiple model or script files, you can only have one access control file in any business network.

Finally, click on the Update button to save the model and script files to the network.

Next, we need to test our business network by creating some participants (in this case Traders), creating an asset (a Commodity), and then using our Trade transaction to change the ownership of the Commodity.

Click the Test tab to get started.

The first thing we should add to our business network is two participants.

Ensure that you have the Trader tab selected on the left, and click Create New Participant in the upper right.

What you can see is the data structure of a Trader participant. We want some easily recognizable data, so delete the code that's there and paste the following:

```
{
  "$class": "com.acmewidgets.mynetwork.Trader",
  "tradeId": "TRADER2",
  "firstName": "Joe",
  "lastName": "Smith"
}
```

Repeat these steps and add a new trader.

```
{
  "$class": "com.acmewidgets.mynetwork.Trader",
  "tradeId": "TRADER1",
  "firstName": "Braun",
  "lastName": "Brelin"
}
```

Now, we'll add a new asset. Click on the assets tab on the far left and create a new asset. Type in the following data for the new asset.

```
{
  "$class": "com.acmewidgets.mynetwork.Commodity",
  "tradingSymbol": "WDG1",
  "description": "A Widget",
  "mainExchange": "WidgetExchange",
  "quantity": 500,
  "owner": "resource:com.acmewidgets.mynetwork.Trader#TRADER1"
}
```

Now that we have both an asset and two traders, let's submit a new transaction to trade the ownership of the asset from trader 1 to trader 2.

Click on the 'submit transaction' button.

Type in the following:

```
{
 "$class": "com.acmewidgets.mynetwork.Trade",
 "commodity": "resource:com.acmewidgets.mynetwork.Commodity#WDG1",
 "newOwner": "resource:com.acmewidgets.mynetwork.Trader#2",
}
```

Now submit the trade. Clicking on the ‘all transactions’ button in the transaction tab wil give you a list of all the transactions on the network. Click on ‘view record’ to view the record for the transaction that you just created and submitted.