CST4025
Blockchain Development
Week 6
Dr Ian Mitchell

1 Introduction

All today's exercises are to be included in week 6 directory. Open up composer playground by going to `https://composer-playground.mybluemix.net`.

2 T4

In the lecture we adapted the trader example available from hyperledger. From the hperledger composer-playground business network page, open up a new network with the following details:

- name: t4
- namespace: org.t4.net
- Admin: admin@t4.net

Access the business network via the composer-playground interface and you should see a similar webpage as displayed in Fig 1. There are 4 files that we

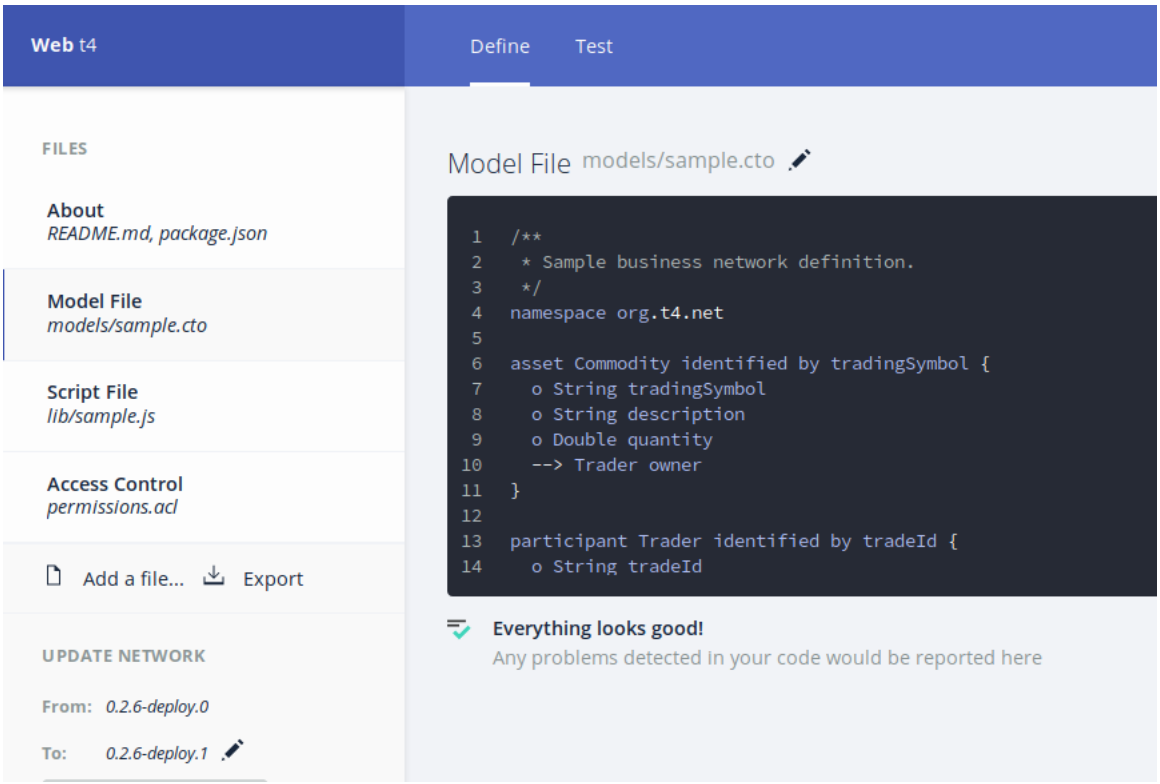

Figure 1: Composer Playground development environment

```
/**
 * Sample business network definition.
 */
namespace org.t4.net

asset Commodity identified by tradingSymbol {
  o String tradingSymbol
  o String description
  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
}
```

Figure 2: CTO code for t4 business network archive

intend to control today; the first file contains some very basic information about the blockchain application you are building, please populate as appropriate.

Complete the following instructions:

1. Click on the Model File tab and enter the code in Fig 2
2. Click on the Script File tab and enter the code in Fig 3
3. Click on the Access Control tab and enter the code in Fig 4
4. Click deploy network and then Test
5. Create two participants with id ‘0001’ and ‘0002’, respectively
6. Create 1 commodity with id ‘0010’ with ownership registered to trader with id, ‘0001’
7. Complete the transaction that transfers the ownership of the commodity, ‘0010’ to a new trader, ‘0002’.
8. Complete the transaction of the same resource, ’0010’, to a participant that does not exist. Does the transaction complete?
9. Write the code using the method exist, prevent the above transaction from occuring and ensure that only existing participants can be the new owner.

```
/**
 * transaction of a commodity from one trader to another
 * This check could be completed with ACL and is an exercise
 * @param {org.t4.net.Trade} trade - the trade to be processed
 * @transaction
 */
async function tradeCommodity(tx) {
  var ns="org.t4.net.";
    tx.commodity.owner=tx.newOwner;
    const commodityRegister = await getAssetRegistry(ns+"Commodity");
    await commodityRegister.update(tx.commodity);
}
```

Figure 3: JavaScript code for t4 business network archive

```
/**
 * Sample access control list.
 */

rule SystemACL {
    description: "System ACL to permit all access"
    participant: "org.hyperledger.composer.system.Particpant"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}

rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}

rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

Figure 4: Access Control code for t4 business network archive

3 Add Method

In this section we are going to Add a new member of staff. Add the following code in Figs 5, 6 and 7 to your Access Control File, Model File and Script File in composer playground, respectively.

Complete the following:

- Create three users, with clerk, consultant and manager status
- Issue new ID and wallets for these three users, call them clerk, consultant and manager
- Enter the system as a manager and test the addStaff function. Does it work? Look for console.log output.
- Enter the system as a clerk and test the addStaff function. Does it work? Look for the error.
- Complete your blockchain by introducing a new function that removes a member of staff from the participant registry, use the lecture notes and hyperledger website to help you.
- Test and evaluate your remove function, are there any restrictions and can they be overcome?

```
/**
 * Sample access control list.
 */
rule CreateTrader{
  description: "Allow traders to add new traders"
  participant: "org.t4.net.Trader"
  operation: ALL
  resource: "org.t4.net.*"
  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
}

rule NetworkAdminUser {
    description: "Grant business network administrators full access to user resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "**"
    action: ALLOW
}

rule NetworkAdminSystem {
    description: "Grant business network administrators full access to system resources"
    participant: "org.hyperledger.composer.system.NetworkAdmin"
    operation: ALL
    resource: "org.hyperledger.composer.system.**"
    action: ALLOW
}
```

Figure 5: Access Control code for allowing create privileges to add staff business network archive

```
/**
 * Sample business network definition.
 */
namespace org.t4.net

enum Grade {
  o manager
  o consultant
  o intern
  o clerk
}

⋮

transaction AddStaff{
  o Trader newTrader
}
```

Figure 6: CTO code for adding Status to business network archive

```
/*
 * transaction to add new member of staff
 * @param {org.t4.net.AddStaff} AddStaff - add new staff
 * @transaction
 */
async function addNewStaff(tx){
  var ns1='org.t4.net';
  var me=getCurrentParticipant();
  if (me.Status === 'manager') {
    return getParticipantRegistry(ns1+'.Trader')
      .then( function(traderRegistry){
            //console.log('me.status value: '+me.Status);
            var factory = getFactory();
            var newStaff = factory.newResource(ns1, 'Trader', tx.newTrader.tradeId);
            console.log('new factory: complete');
            newStaff=tx.newTrader;
          console.log('new trader:'+newStaff.tradeId);
            return traderRegistry.add(newStaff);
    })
    }
    else
    {
          throw new Error("You have insufficient privileges to add a member of staff");
    }
}
```

Figure 7: JavaScript code for Adding member of staff to business network archive