## Starting Hyperledger Fabric

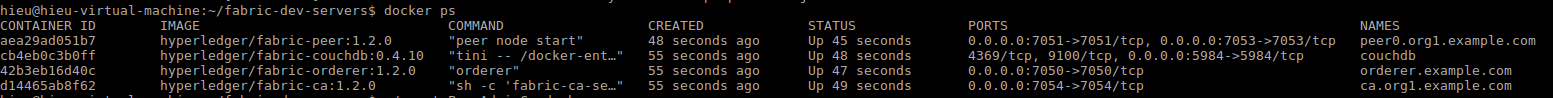
git clone <https://github.com/IBM/BlockchainSmartContractTrading-CompositeJourney.git>

cd fabric-dev-servers

./downloadFabric.sh

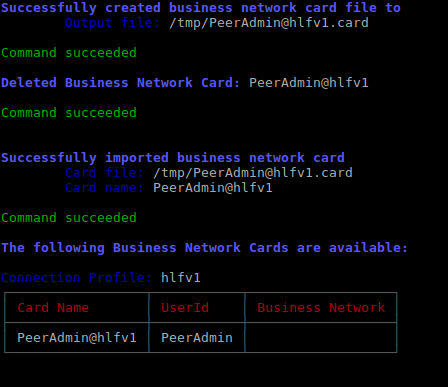
./startFabric.sh

Create network peer -> docker ps



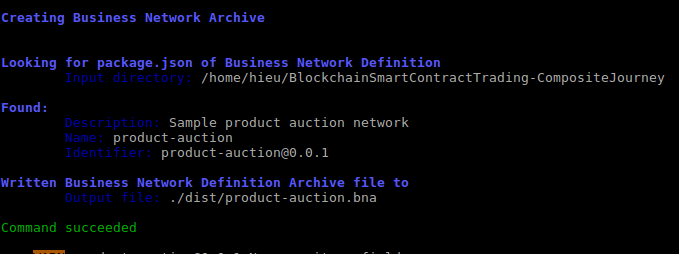
NOTE : Clear all existing card : composer card delete –card <card name>

./createPeerAdminCard.sh

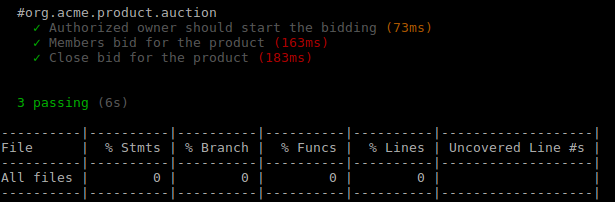


## Generate the Business Network Archive (BNA)

./ BlockchainSmartContractTrading-CompositeJourney-> npm install



Npm test

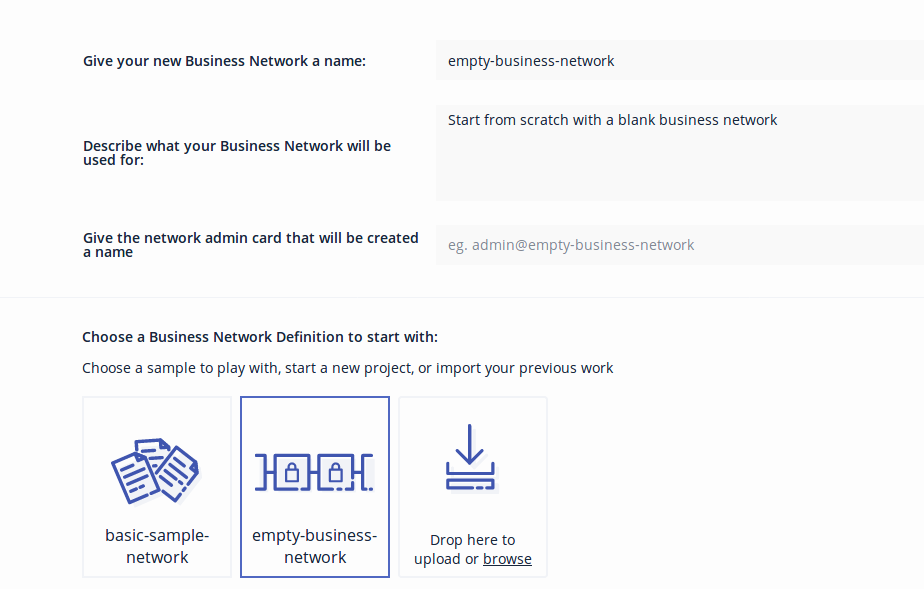


**Deploy the Business Network Archive using Composer Playground**

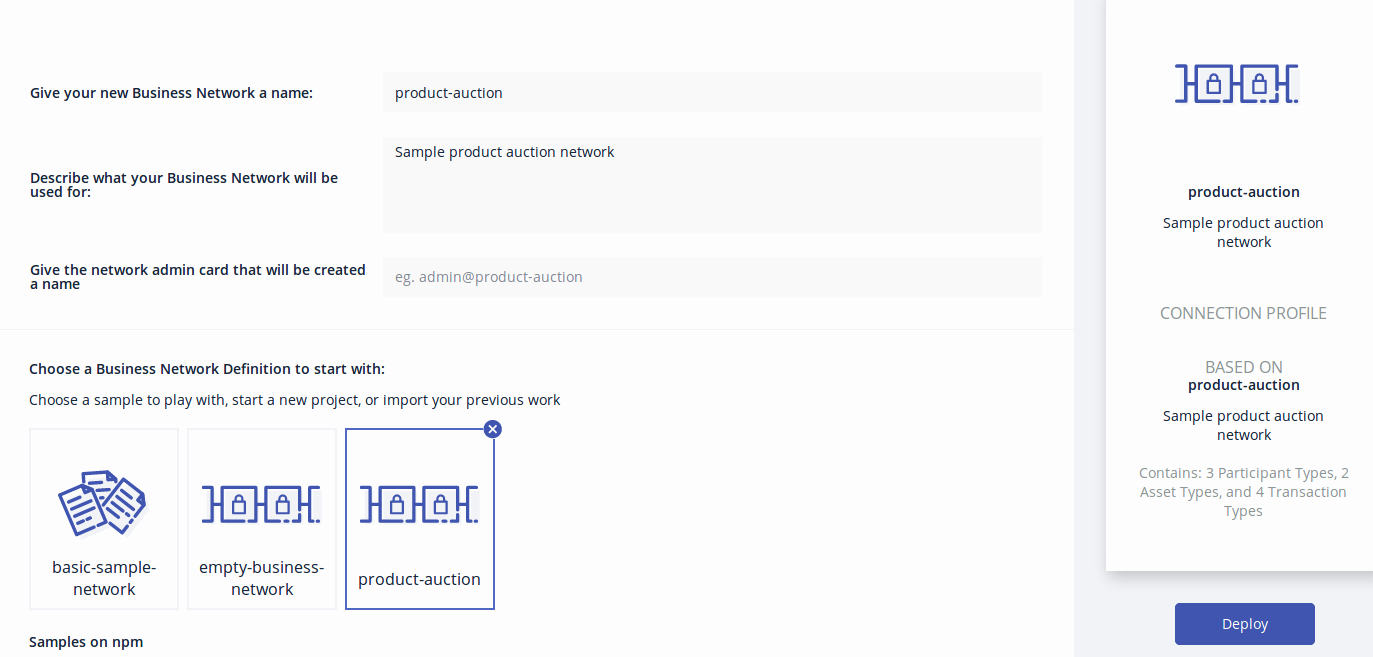
Open [Composer Playground](http://composer-playground.mybluemix.net/) : <http://composer-playground.mybluemix.net/login>

Deploy a new business network

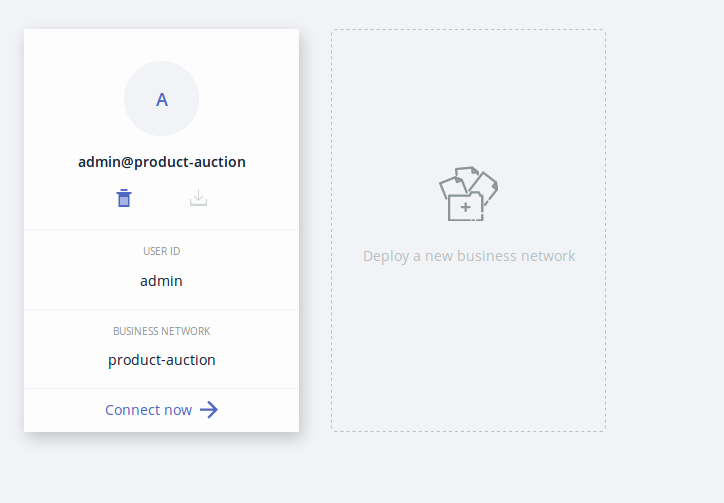
Drop here to Upload or Browse



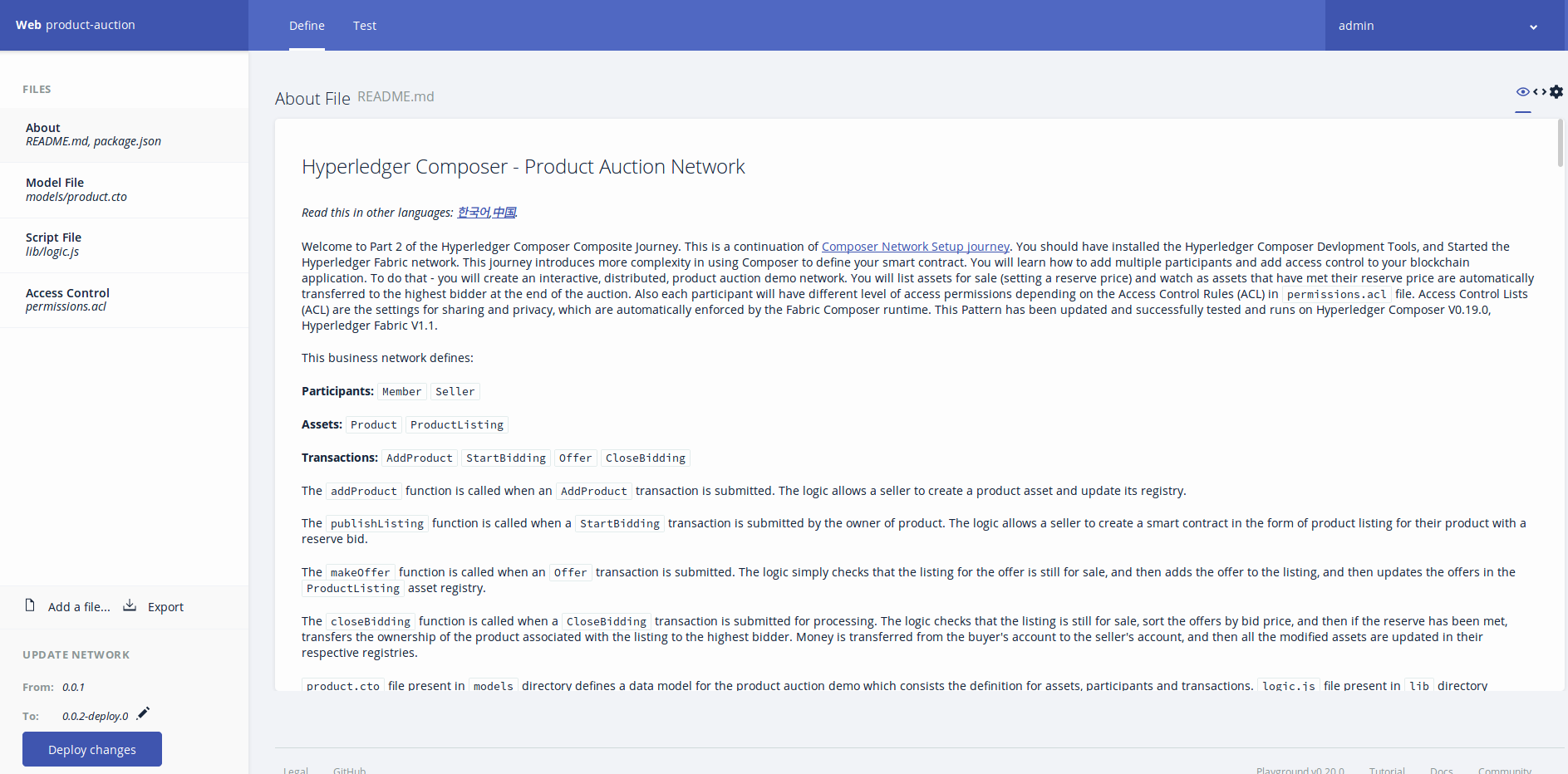
./dist/product-auction.bna -> Open



Connect now



You will see the following:



To test this Business Network Definition in the **Test** tab:

In the **Seller** participant registry, create a new participant. Make sure you click on the **Seller** tab on the far left-hand side.

{

"$class": "org.acme.product.auction.Seller",

"organisation": "ACME",

"email": "auction@acme.org",

"balance": 100,

"products": []

}

In the **Member** participant registry, create two participants. Again, click on the **Member** tab on the far left-hand side.

{

"$class": "org.acme.product.auction.Member",

"firstName": "Amy",

"lastName": "Williams",

"email": "memberA@acme.org",

"balance": 1000,

"products": []

}

{

"$class": "org.acme.product.auction.Member",

"firstName": "Billy",

"lastName": "Thompson",

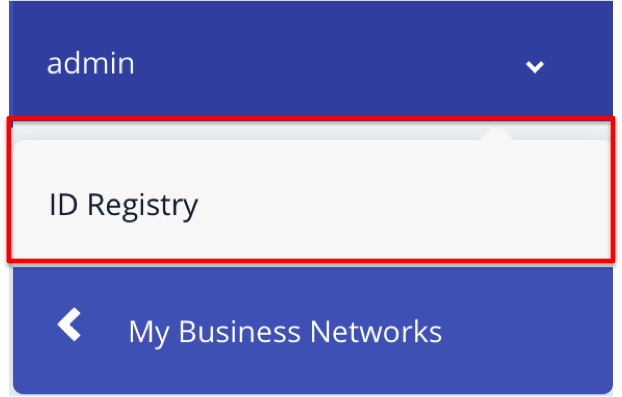
"email": "memberB@acme.org",

"balance": 1000,

"products": []

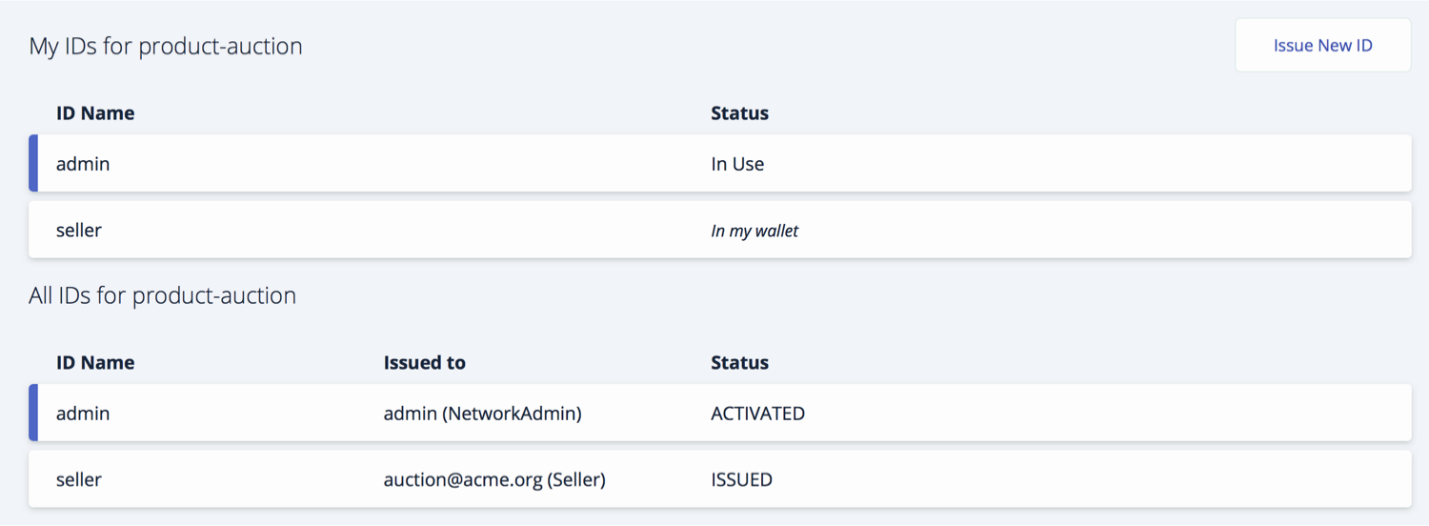
}

Now we are ready to add **Access Control**. Do this by first clicking on the admin tab to issue **new ids** to the participants. Note: the ids are automatically added to the wallet.

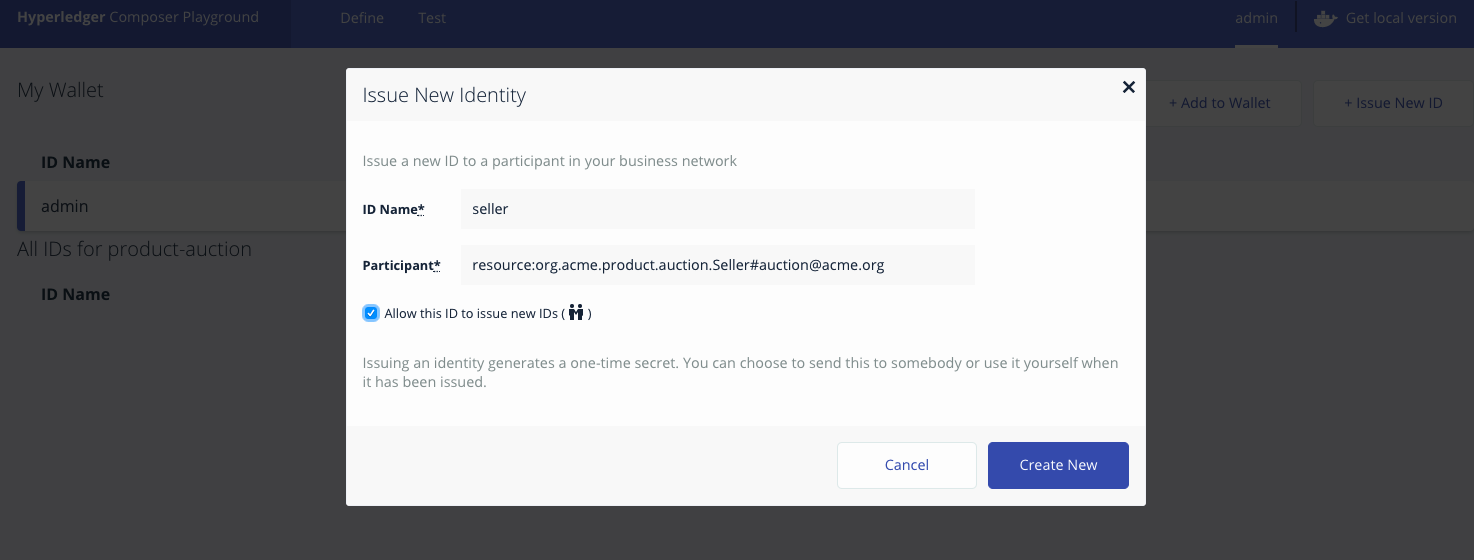
Select Admin-> ID Registry You will see the following:

Please follow the instructions as shown in the images below:

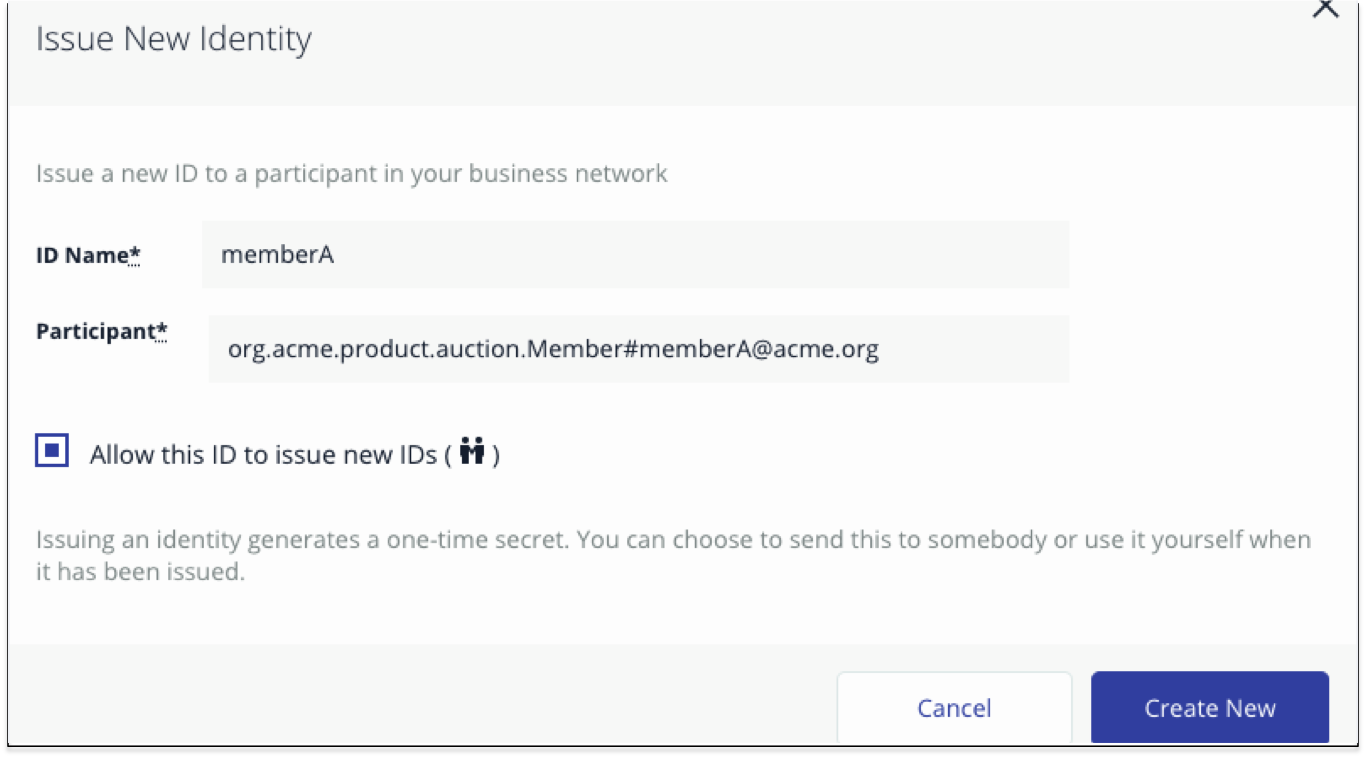
Click on **Issue New ID** button on upper-right hand side - the follow pop-up will appear:



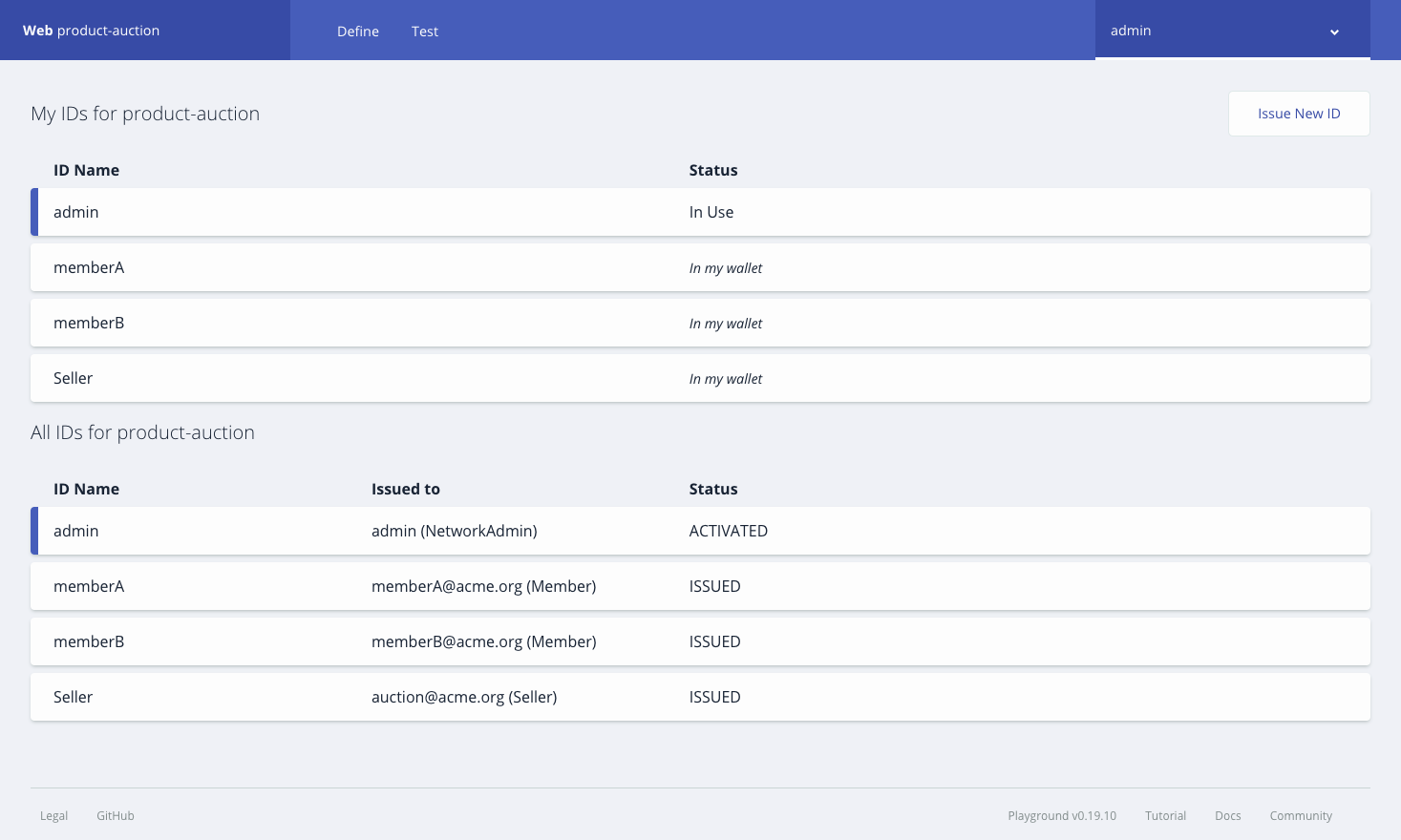
Enter **Seller** for ID Name. Then enter "org.acme.product.auction.Seller#[auction@acme.org](mailto:auction@acme.org)" in the participant field. **Note**: there is case-sensitivity wrt the name **Seller**. If you have it capitilized as a participant when you added participants under the **Test** page - you must match it here the way you enter it in the Participant field. Ensure you have checked the **Allow this ID to issue new IDs** checkbox. Select the **Create New** button.



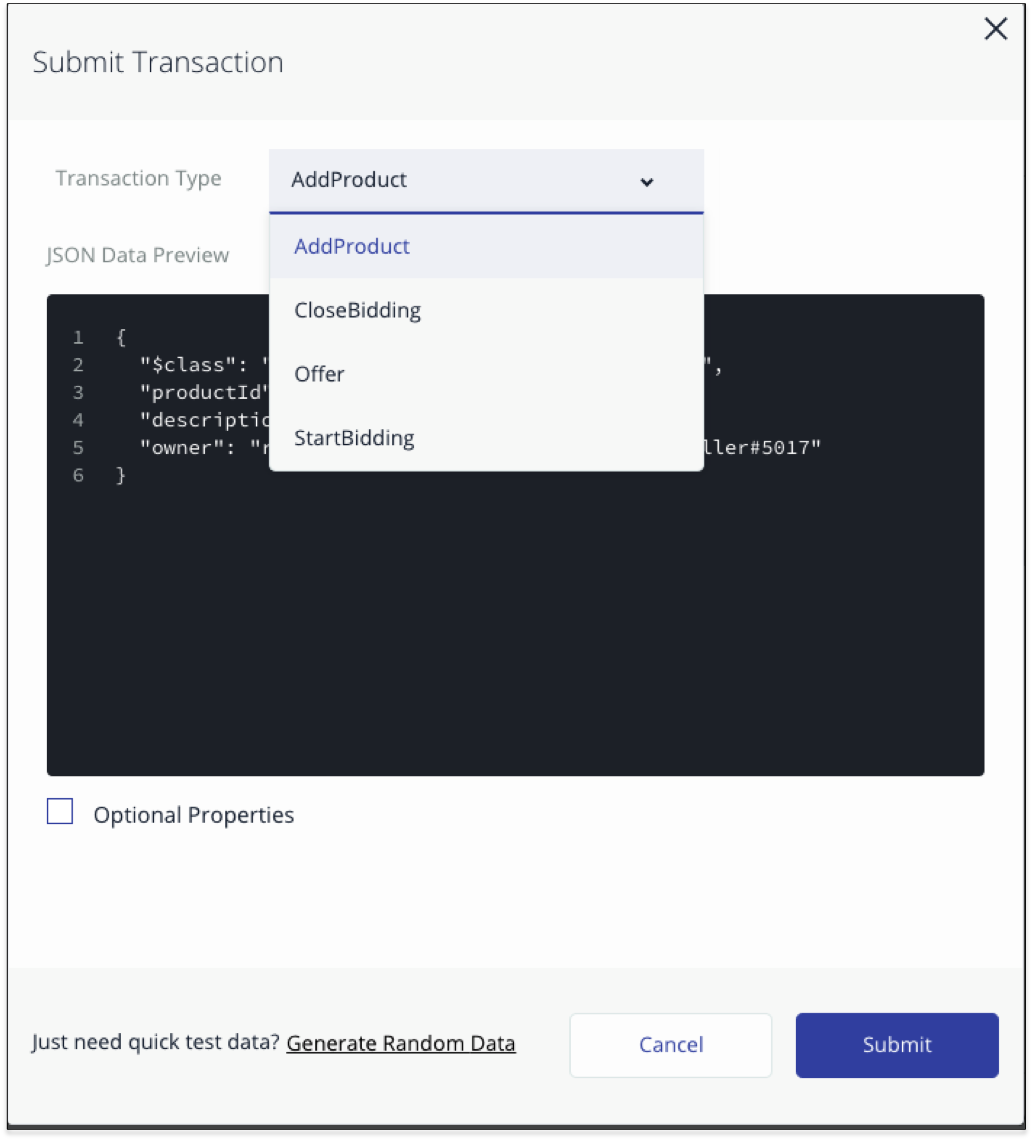
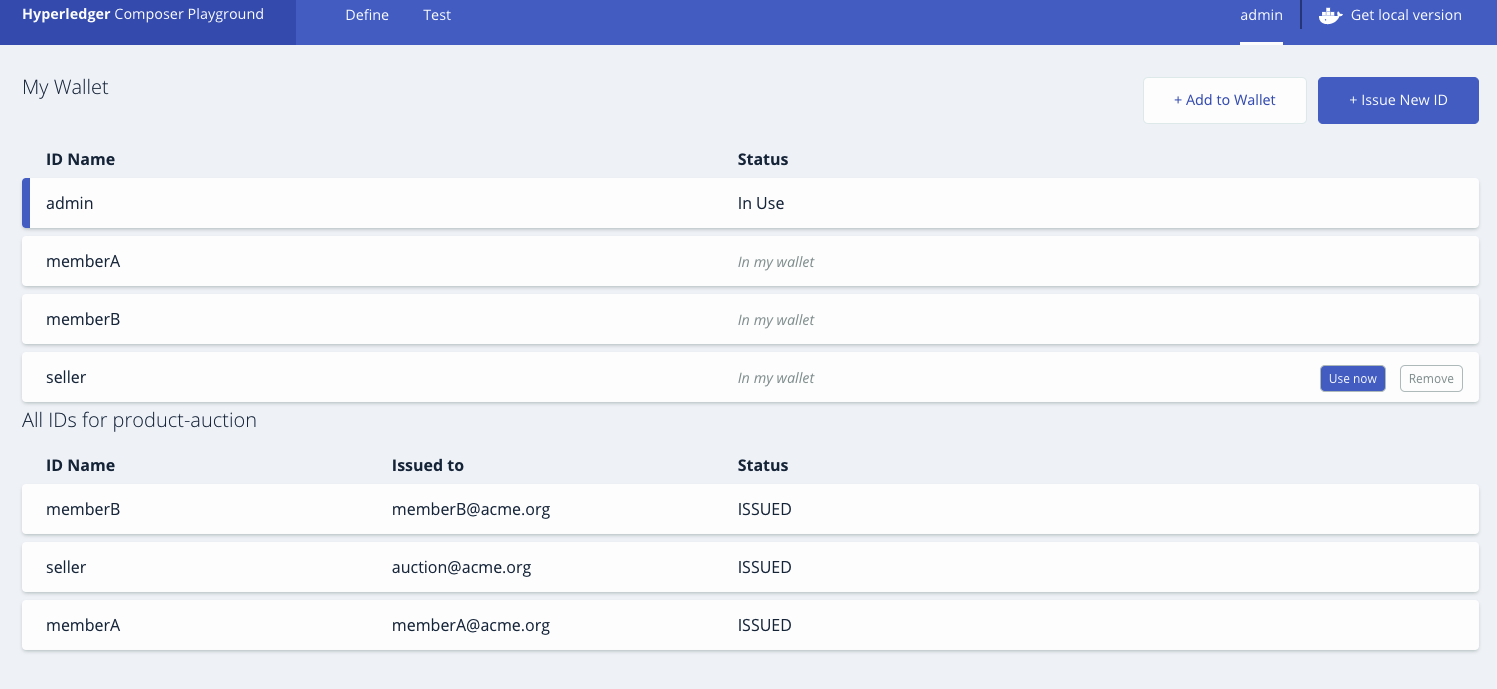
Now issue IDs for MemberA and MemberB (example of issuing ID for MemberA in the graphic below - duplicate process for MemberB). First, again, select **Issue New ID**. Then complete the fields and check the checkbox. Select **Create New**.



Once you complete those steps - your screen should appear as follows:



The Wallet tab is pictured in the image below. Select the **seller id** from **Wallet tab** tab (as demonstrated in the graphic below). Select the 'use now' button. Note the status of the **Seller** id is now **In Use**. Now click on the **test tab** to perform **AddProduct** and **StartBidding** transactions.



Now click on **Submit Transaction** button and select **AddProduct** transaction from the dropdown, to create a product for the seller.

To create a product listing for the above product, submit **StartBidding** transaction. Again, select **Submit Transaction** button and then select **Start Bidding** transaction from the dropdown.

{

"$class": "org.acme.product.auction.AddProduct",

"productId": "p1",

"description": "Sample Product",

"owner": "resource:org.acme.product.auction.Seller#auction@acme.org"

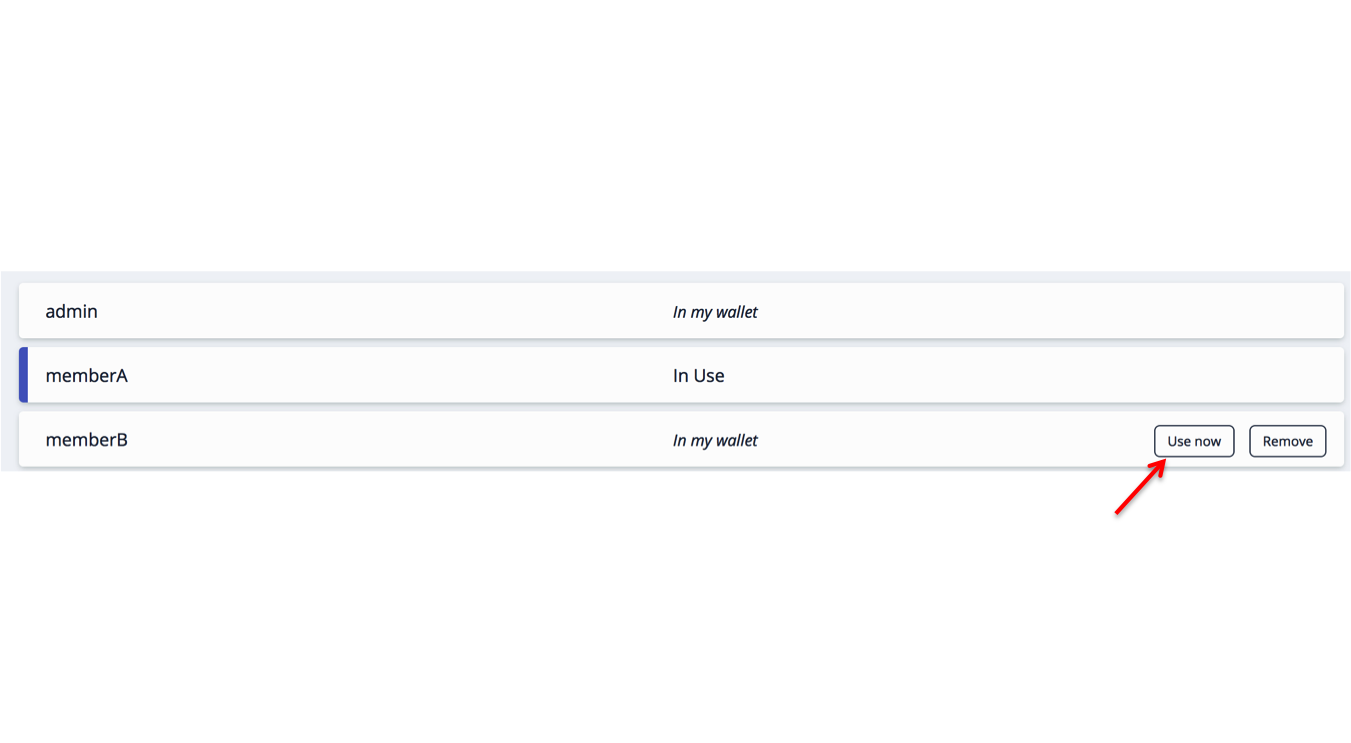
}

elemnet

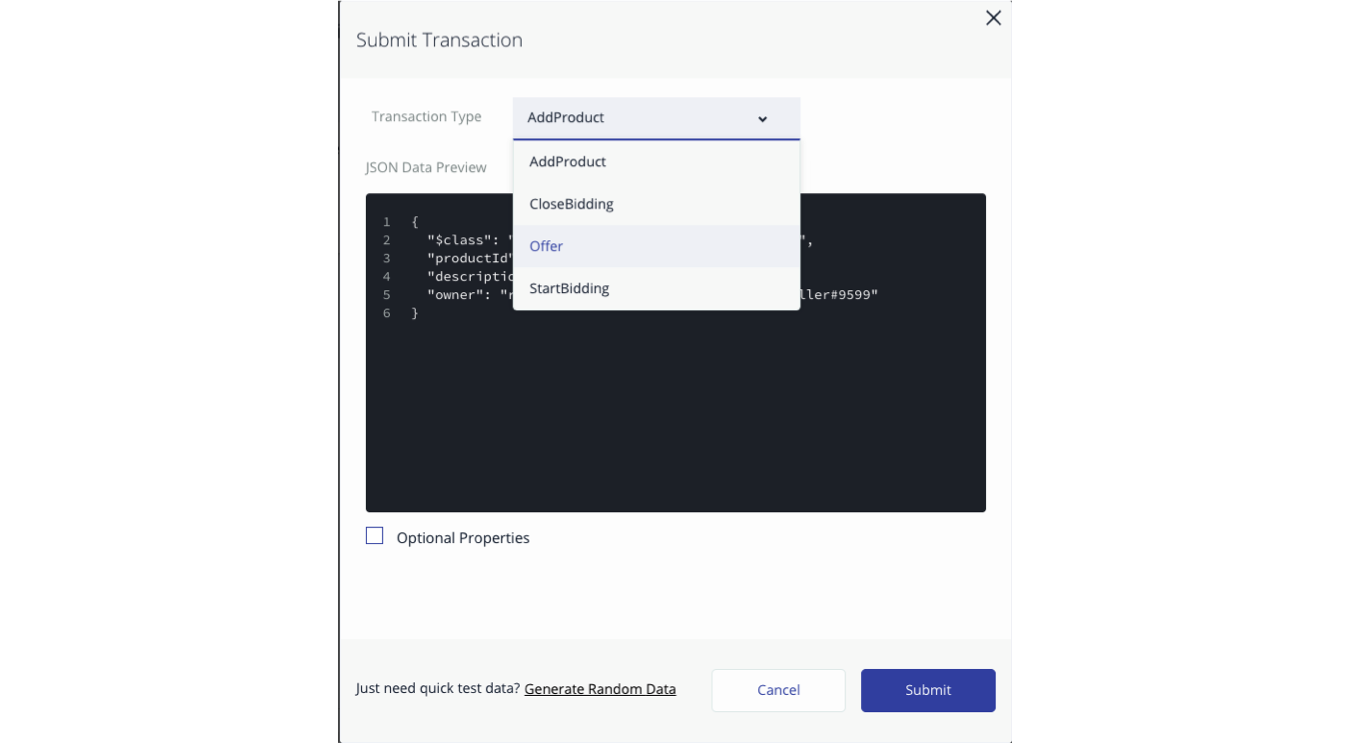
You've just listed **Sample Product - P1** for auction, with a reserve price of $50. A listing has been created in **ProductListing** registry for the product with **FOR\_SALE** state.

Now Member participants can submit **Offer** transactions to bid on a product listing.

For each **member id,** select the user id from the tab on the upper right hand-side that probably says **Seller** at the moment. Select MemberA on the left hand side and then **use now** as is demonstrated in the graphic below.



To submit an **Offer** transaction select the **test tab** and click on **Submit Transaction** button. Select **offer** from the drop down.



{

"$class": "org.acme.product.auction.Offer",

"bidPrice": 50,

"listing": "resource:org.acme.product.auction.ProductListing#l1",

"member": "resource:org.acme.product.auction.Member#memberA@acme.org"

}

Repeat the process for MemberB. Remember to select 'use now' for **memberB** in the registry similar to what you did for **memberA**.

{

"$class": "org.acme.product.auction.Offer",

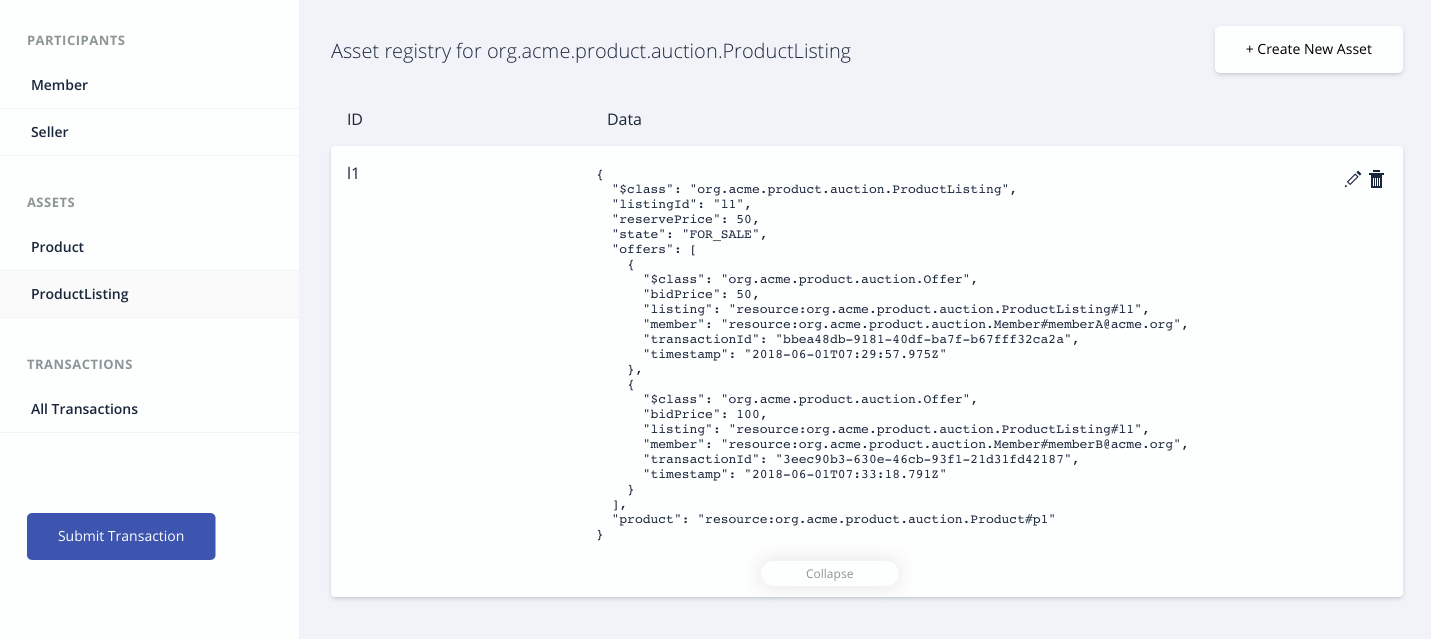
"bidPrice": 100,

"listing": "resource:org.acme.product.auction.ProductListing#l1",

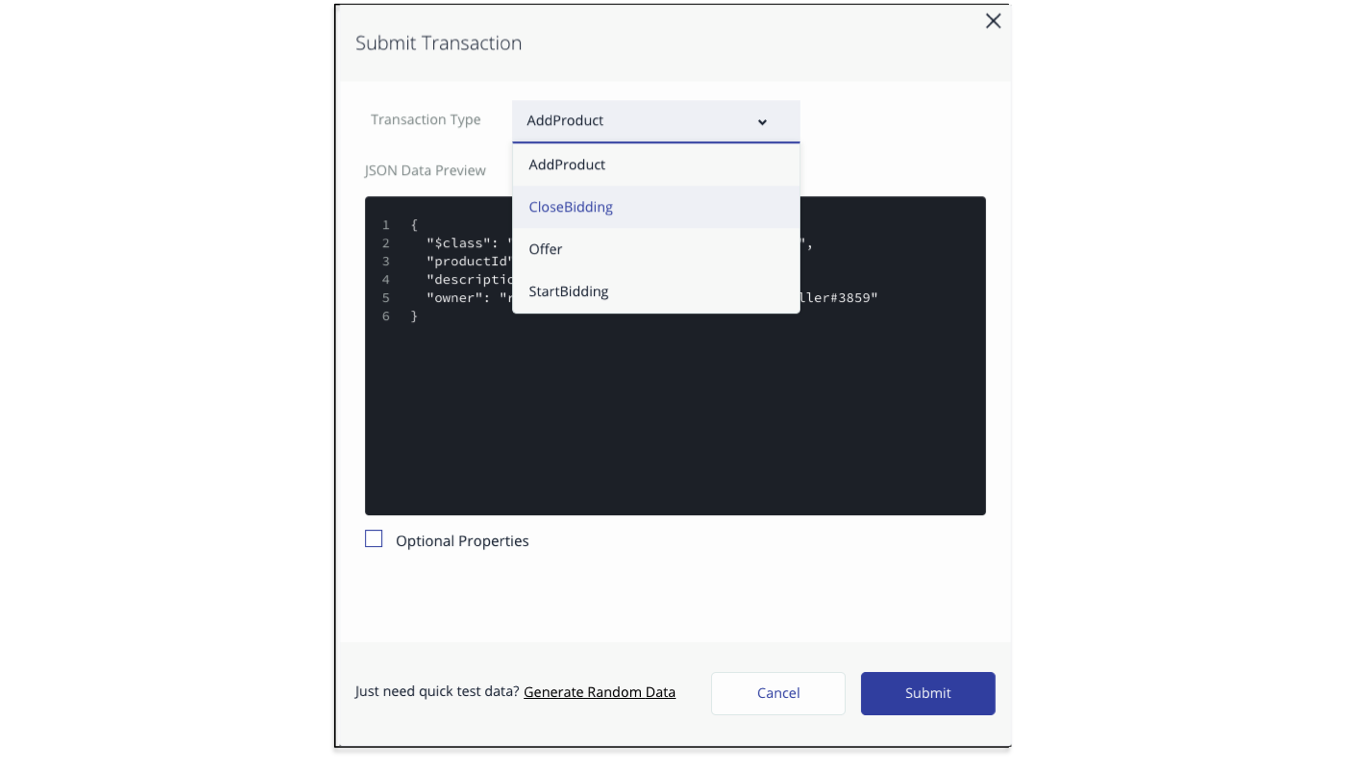
"member": "resource:org.acme.product.auction.Member#memberB@acme.org"

}

You can check the **ProductListing** registry, to view all the bids for the product.



Now again select the **seller id** from the **Wallet tab** tab. Set it to **use now**. Click on **test tab** and then end the auction by submitting a **CloseBidding** transaction for the listing.

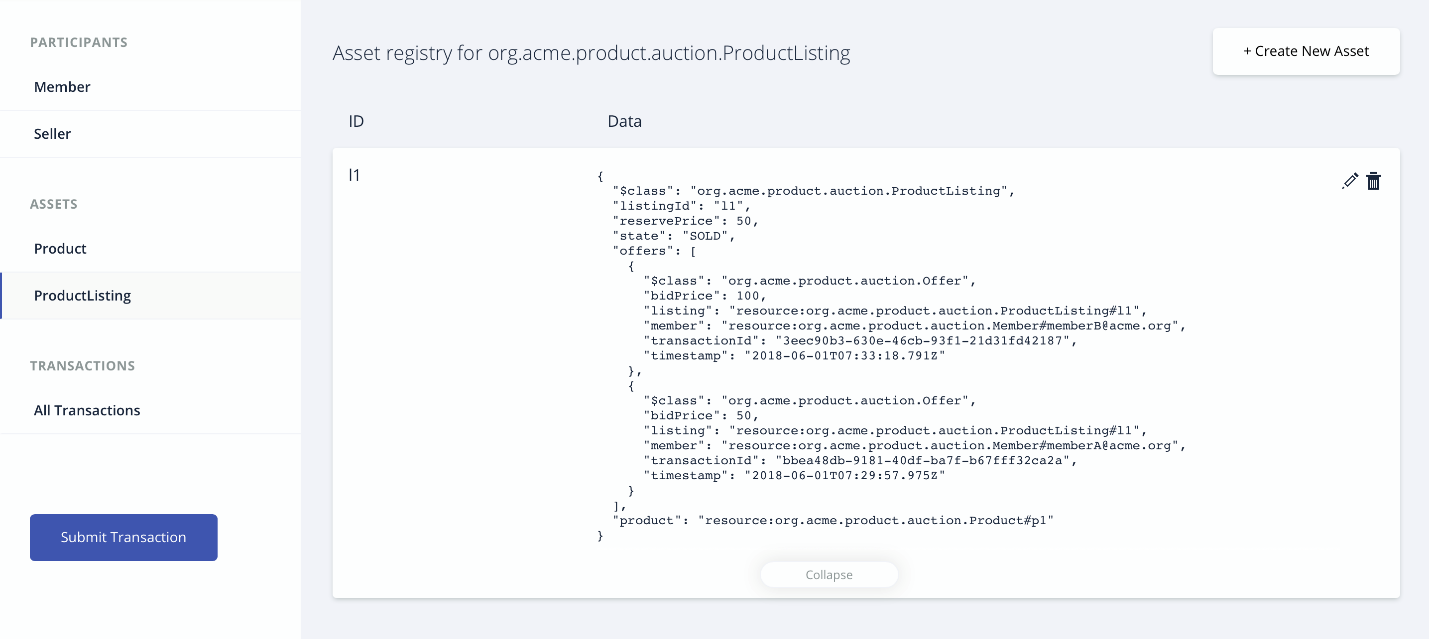


{

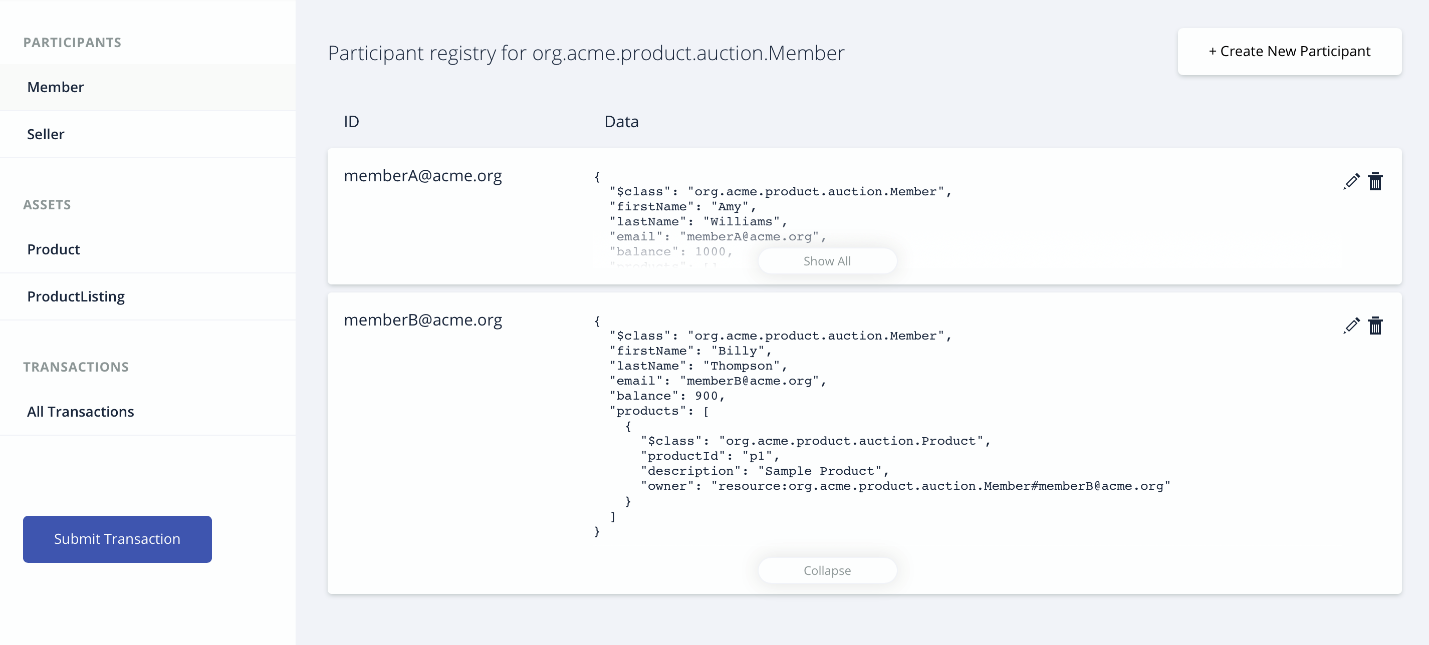
"$class": "org.acme.product.auction.CloseBidding",

"listing": "resource:org.acme.product.auction.ProductListing#l1"

}

You can check the state of the ProductListing with **l1** is **SOLD**. 

Click on the **Member** asset registry to verify the updated balance for buyer and seller. The product is added to the product list of the buyer **memberB@acme.org**. Note the balance of MemberB is $900.



## Deploy the Business Network Archive on Hyperledger Composer running locally

cd dist

composer network install --card PeerAdmin@hlfv1 --archiveFile product-auction.bna

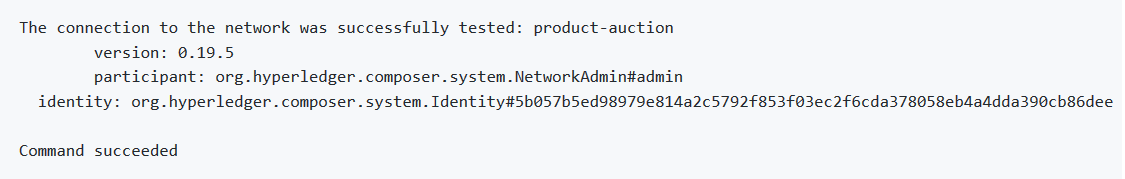
composer network start --networkName product-auction --networkVersion 0.0.1 --networkAdmin admin --networkAdminEnrollSecret adminpw --card PeerAdmin@hlfv1 --file networkadmin.card

composer card import --file networkadmin.card

You can verify that the network has been deployed by typing:

composer network ping --card admin@product-auction

You should see the the output as follows:



To create the REST API we need to launch the **composer-rest-server** and tell it how to connect to our deployed business network. Now launch the server by changing directory to the product-auction folder and type:

cd ..

composer-rest-server

Answer the questions posed at startup. These allow the composer-rest-server to connect to Hyperledger Fabric and configure how the REST API is generated.

* Enter **admin@product-auction** as the card name.
* Select **never use namespaces** when asked whether to use namespaces in the generated API.
* Select **No** when asked whether to secure the generated API.
* Select **Yes** when asked whether to enable event publication.
* Select **No** when asked whether to enable TLS security.

**Test REST API**

If the composer-rest-server started successfully you should see these two lines are output:

Web server listening at: http://localhost:3000

Browse your REST API at http://localhost:3000/explorer

Open a web browser and navigate to <http://localhost:3000/explorer>