

Chapter 7: Seller Support and User Experience

Learning Bluemix & Blockchain

Bob Dill, IBM Distinguished Engineer, CTO Global Technical Sales
David Smits, Senior Certified Architect, IBM Blockchain



The Plan: 30 minute Chapters with an hour or two of practice

Chapter 1:	What is Blockchain? Concept and Architecture overview
Chapter 2:	What's the story we're going to build
Chapter 2.1:	Architecture for the Story
Chapter 3:	Set up local HyperLedger V1 development environment
Chapter 4:	Build and test the network
Chapter 5:	Administration User Experience
Chapter 6:	Buyer Support and User Experience
Chapter 7:	Seller Support and User Experience
Chapter 8:	Shipper Support and User Experience
Chapter 9:	Provider Support and User Experience
Chapter 10:	Finance Company Support and User Experience
Chapter 11:	Combining for Demonstration
Chapter 12:	Events and Automating for Demonstration



A couple of starting comments

- Each time you start, or restart docker or your docker containers, they start up in their original, unpopulated state. This means that they have no data already in the system. This happens when you run the `./startup.sh` —or— `./buildAndDeploy.sh` scripts.
- The tutorial has an 'autoload' feature as part of the admin menu at the top of the screen. This will automatically create 20 members, 9 orders and the item file used during the 'create new order' process for the buyer.
- This command needs to be run if you want to prepopulate your systems with some data. We will look at adding transactions to this autoload process in the final chapter.
- Autoload will display its status, shown to the right, via a web socket channel.

```
connected
[6] Cloud Nine Software, Inc successfully added
[10] Software Solutions, Inc successfully added
[18] dummy shipper successfully added
[15] Fast Eddy, Inc successfully added
[11] Hybrid Cloud Designs, Inc successfully added
[14] The Education Game, Inc successfully added
[0] PC Hardware, Inc successfully added
[4] 2nd Life Systems, Inc successfully added
[1] Innovative Solutions, Inc successfully added
[9] Virtual Paper, Inc successfully added
[2] The i-Series Experts, Inc successfully added
[16] The Overnight Experts, Inc successfully added
[21] The App Store, Inc successfully added
[20] UPS Systems, Inc successfully added
[8] NonStop, Inc successfully added
[13] Kid Friendly Learning, Inc successfully added
[12] Born On The Cloud, Inc successfully added
[5] The Cognitive Advantage, Inc successfully added
[17] PC Hardware Now, Inc successfully added
[3] Cooling Systems R Us, Inc successfully added
[19] dummy provider successfully added
[7] Office Experts, Inc successfully added
[22] The Global Financier successfully added
loadTransaction: order 008 successfully added
loadTransaction: order 006 successfully added
loadTransaction: order 009 successfully added
loadTransaction: order 005 successfully added
loadTransaction: order 003 successfully added
loadTransaction: order 004 successfully added
loadTransaction: order 007 successfully added
loadTransaction: order 001 successfully added
loadTransaction: order 002 successfully added
```



You can check the status of all orders in the system via the admin interface we built in Chapter 5

Zero to Blockchain Tutorial

Select Language ▾

Admin ▾

Roles ▾

Home

Zero To Blockchain Chapter 10

Network Profile Management

API	Parameters
create Profile	profile object
delete profile	profile name
get all connection profiles	(none)
get a specific network connection profile	profile name

Business Network Management

API	Parameters
deploy a network	network archive file, options
install new a network	network archive file, options
start an installed network	network name, options
list the deployed business networks	(none)
touch a network, check compatibility	business network name
take a business network off line	business network name
update an existing business network	business network name, archive file

Resource Management

API	Parameters
list members of a registry	Buyer ▾
List Assets in the registry	(none)
Add Member	Co Name, id, Type
Remove Member	Buyer ▾
getMemberSecret	Buyer ▾

Result

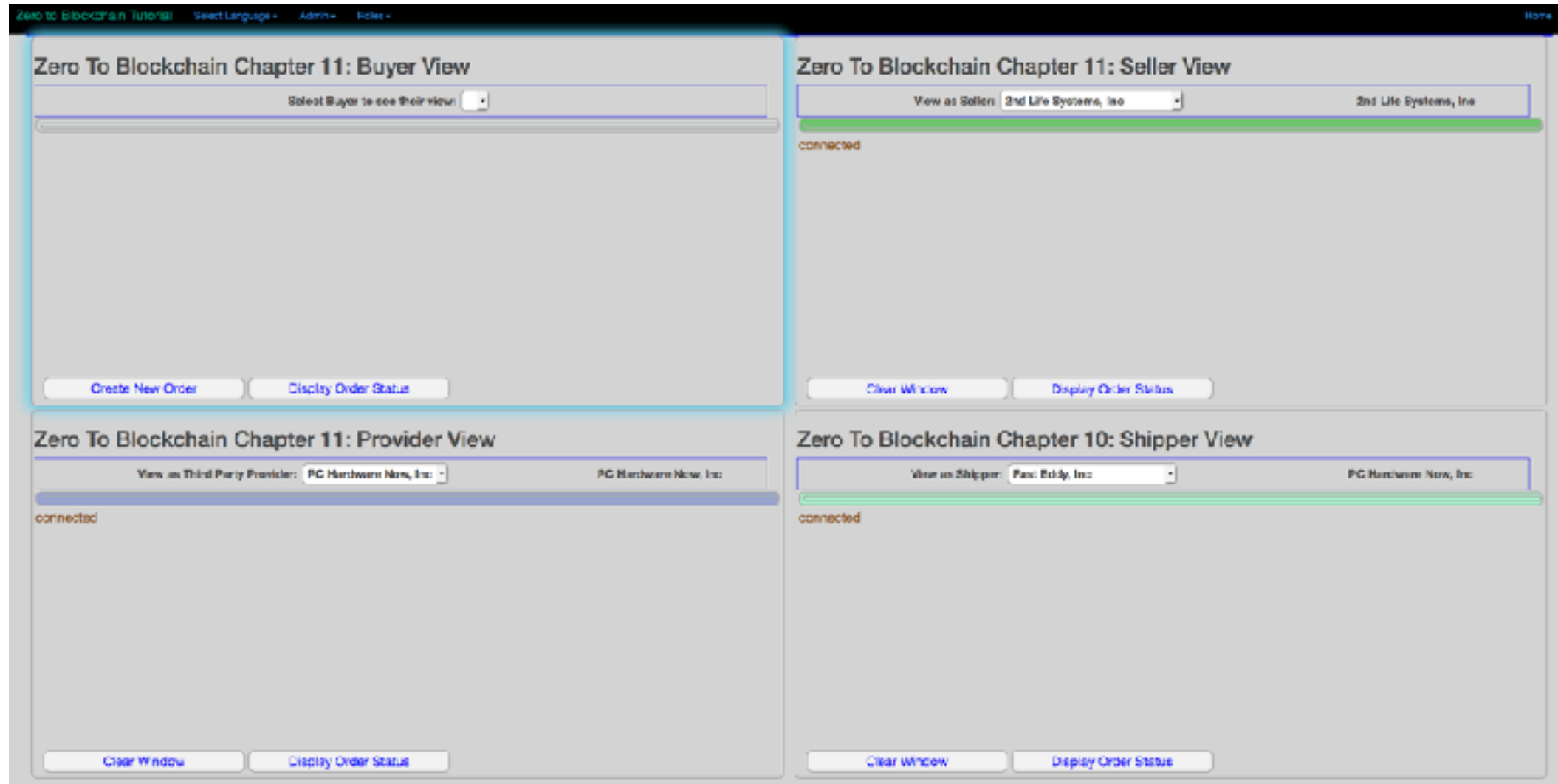
Registry List

Network update results: success

Order Number	Created	Status	Buyer/Seller	Amount
001	2017-09-25T11:59:04.793Z	Order Created	yes@softwaresolutionsinc.com bob@pchardwareinc.com	\$9125.00
002	2017-09-25T11:59:04.795Z	Order Created	yes@softwaresolutionsinc.com jojo@innovativesolutionsinc.com	\$6055.00
003	2017-09-25T11:59:03.770Z	Order Created	yes@softwaresolutionsinc.com Erin@2ndlifesystemsinc.com	\$38800.00
004	2017-09-25T11:59:03.786Z	Order Created	yes@softwaresolutionsinc.com alesha@cloudninesoftwareinc.com	\$9125.00
005	2017-09-25T11:59:03.748Z	Order Created	yes@softwaresolutionsinc.com joshua@officeexpertsinc.com	\$6055.00
006	2017-09-25T11:59:01.937Z	Order Created	eric@bornonthecloudinc.com Erin@2ndlifesystemsinc.com	\$38800.00
007	2017-09-25T11:59:04.714Z	Order Created	eric@bornonthecloudinc.com bob@pchardwareinc.com	\$9125.00
008	2017-09-25T11:58:56.406Z	Order Created	eric@bornonthecloudinc.com jojo@innovativesolutionsinc.com	\$6055.00
009	2017-09-25T11:59:03.746Z	Order Created	eric@bornonthecloudinc.com Erin@2ndlifesystemsinc.com	\$38800.00

Corporation

Over the next three lessons, we will build each of the panels in this picture. Today, we're focused on the top right: Seller



In Chapter 6, we created the buyer view, which allowed us to create orders, display orders, and selectively update the status of orders. Today, we'll create the user experience for the Seller, to allow them to selectively update the status of an order.

Zero To Blockchain TutorialSelect LanguageArticlesNotesHome

Zero To Blockchain Chapter 11: Buyer View

Select Buyer to see their view:abbykiddmrandyano.comKid Friendly Learning, Inc

Order #	Status	Total	Seller: 2nd Life Systems, Inc
abbykiddmrandyano.com160949966064	Order Purchased: 2017-09-26T12:00:00Z	\$1265.00	No ActionExecute
Item Number	Description	Quantity	Price
1	Macbook Pro - 6Gb, 1Tb	2	\$2570.00
2	4K Monitor - 42	1	\$1985.00

Zero To Blockchain Chapter 11: Provider View

View as Third Party Provider:PC Hardware Now, IncPC Hardware Now, Inc

connected

Clear WindowDisplay Order Status

Zero To Blockchain Chapter 11: Seller View

View as Seller:2nd Life Systems, Inc2nd Life Systems, Inc

Order #	Status	Total	Buyer: 2nd Life Systems, Inc
006	Order Purchased: 2017-09-26T12:00:00Z	\$35800.00	Order From Supplier:PC Hardware Now, IncExecute
Item Number	Description	Quantity	Price
3	Larson's Third Step: Math 2016-2017	15	\$7500.00
4	Larson's Third Step: Math 2016-2017	20	\$31300.00

Order #	Status	Total	Buyer: Kid Friendly Learning, Inc
abbykiddmrandyano.com160949966064	Order Purchased: 2017-09-26T12:00:00Z	\$4255.00	No ActionExecute
Item Number	Description	Quantity	Price
1	Macbook Pro - 6Gb, 1Tb	2	\$2570.00
2	4K Monitor - 42	1	\$1685.00

order 006 successfully updated to Order From Supplier

Zero To Blockchain Chapter 10: Shipper View

View as Shipper:Fast Eddy, IncPC Hardware Now, Inc

connected

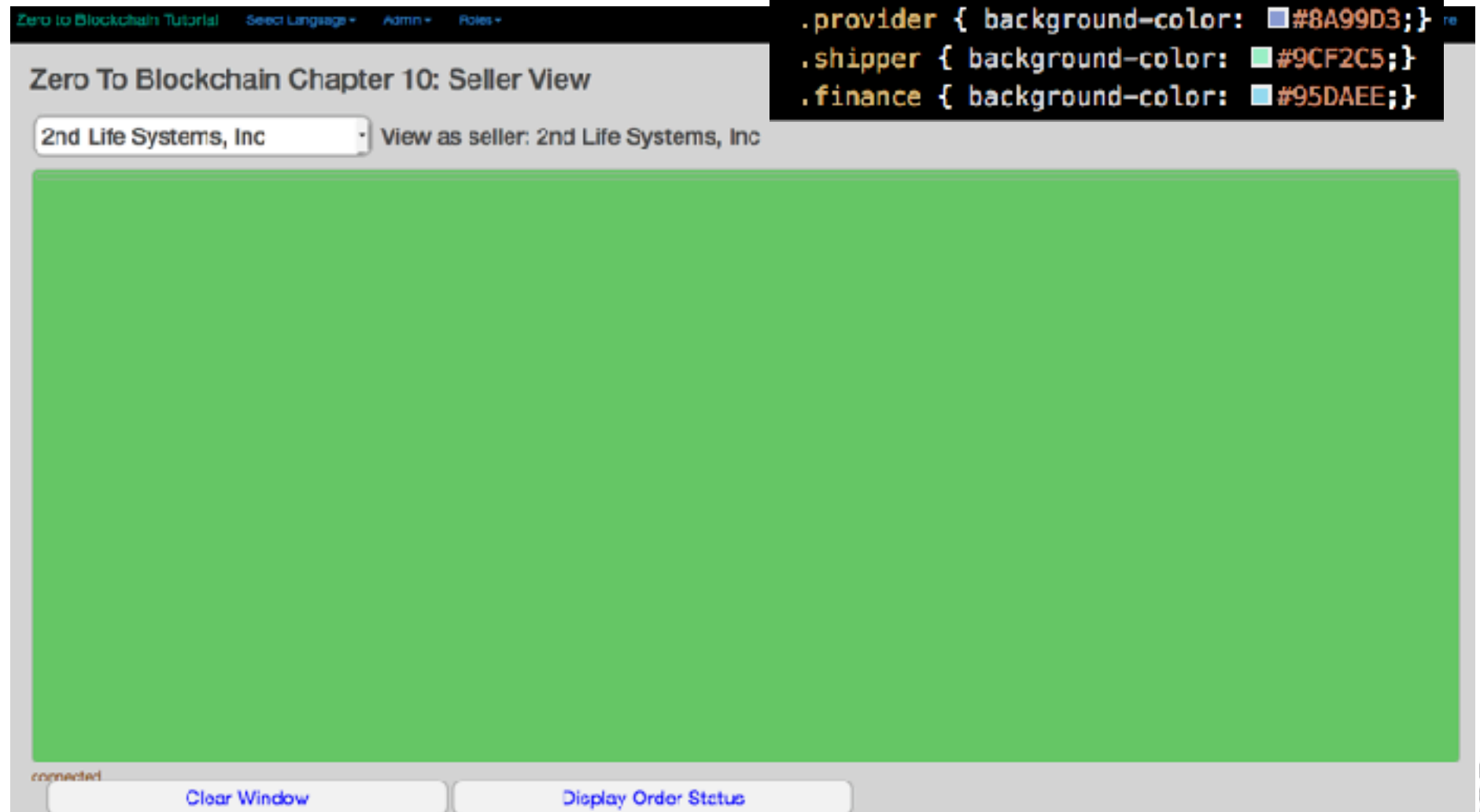
Clear WindowDisplay Order Status



You'll note that the Seller view has a green background. You can specify the color in the `pageStyles.css` file. This is done so that when all 4 roles are on the same browser page, it's easier to differentiate between them.

- We will reuse almost all of the code from the Buyer user experience to create the seller user experience.

```
.buyer { background-color: #CBCFD0; }  
.seller { background-color: #5DC15D; }  
.provider { background-color: #8A99D3; }  
.shipper { background-color: #9CF2C5; }  
.finance { background-color: #95DAEE; }
```



Zero To Blockchain Tutorial Select Language Admin Roles

Zero To Blockchain Chapter 10: Seller View

2nd Life Systems, Inc View as seller: 2nd Life Systems, Inc

connected

Clear Window Display Order Status

We built a query.qry file in the last session, which has the following query in it:

```
query selectOrders {  
  description: "Select all Orders"  
  statement:  
    | SELECT org.acme.Z2BTestNetwork.Order  
}
```

- How does Hyperledger-Composer use the same query for different people and roles? Through the ACL definitions.
- We will do the same thing for the Seller, which will both enable access (if we do nothing, the Seller has absolutely no access) and also limit access to specific transactions.

```
rule BuyerACLCreate {  
  description: "Enable Buyers to execute all actions on an Order"  
  participant(m): "org.acme.Z2BTestNetwork.Buyer"  
  operation: READ, CREATE, UPDATE  
  resource(v): "org.acme.Z2BTestNetwork.**"  
  transaction(tx): "org.acme.Z2BTestNetwork.CreateOrder"  
  condition: (v.buyer.buyerID == m.getIdentifier())  
  action: ALLOW  
}  
  
rule BuyerACLBuy {  
  description: "Enable a Buyer to update an Order from Create"  
  participant(m): "org.acme.Z2BTestNetwork.Buyer"  
  operation: READ, CREATE, UPDATE  
  resource(v): "org.acme.Z2BTestNetwork.**"  
  transaction(tx): "org.acme.Z2BTestNetwork.Buy"  
  condition: (v.buyer.buyerID == m.getIdentifier())  
  action: ALLOW  
}  
  
rule BuyerACLCancel {  
  description: "Enable a Buyer to CANCEL an Order"  
  participant(m): "org.acme.Z2BTestNetwork.Buyer"  
  operation: READ, CREATE, UPDATE, DELETE  
  resource(v): "org.acme.Z2BTestNetwork.**"  
  transaction(tx): "org.acme.Z2BTestNetwork.OrderCancel"  
  condition: (v.buyer.buyerID == m.getIdentifier())  
  action: ALLOW  
}  
  
rule BuyerACLDispute {  
  description: "Enable a Buyer to raise a DISPUTE on an Order"  
  participant(m): "org.acme.Z2BTestNetwork.Buyer"  
  operation: READ, CREATE, UPDATE, DELETE  
  resource(v): "org.acme.Z2BTestNetwork.**"  
  transaction(tx): "org.acme.Z2BTestNetwork.Dispute"  
  condition: (v.buyer.buyerID == m.getIdentifier())  
  action: ALLOW  
}
```


In Chapter 6, we created the OrderAction function, which allowed us to process order status changes for the Buyer. We will extend the set of options in that function to include the seller.

Buyer capability:

```
case 'Purchase':
  console.log('Purchase entered');
  updateOrder = factory.newTransaction(NS, 'Buy');
  updateOrder.buyer = factory.newRelationship(NS, 'Buyer', order.buyer.$identifier);
  updateOrder.seller = factory.newRelationship(NS, 'Seller', order.seller.$identifier);
  break;
case 'Order From Supplier':
  console.log('Order from Supplier entered for "order.orderNumber=" inbound id: "%_userID%" with c');
  updateOrder = factory.newTransaction(NS, 'OrderFromSupplier');
  updateOrder.provider = factory.newRelationship(NS, 'Provider', req.body.provider);
  updateOrder.seller = factory.newRelationship(NS, 'Seller', order.seller.$identifier);
  break;
case 'Resolve':
  console.log('Resolve entered');
  updateOrder = factory.newTransaction(NS, 'Resolve');
  updateOrder.buyer = factory.newRelationship(NS, 'Buyer', order.buyer.$identifier);
  updateOrder.shipper = factory.newRelationship(NS, 'Shipper', order.shipper.$identifier);
  updateOrder.provider = factory.newRelationship(NS, 'Provider', order.provider.$identifier);
  updateOrder.seller = factory.newRelationship(NS, 'Seller', order.seller.$identifier);
  updateOrder.financeCo = factory.newRelationship(NS, 'FinanceCo', financeCoID);
  updateOrder.resolve = req.body.reason;
  break;
case 'Dispute':
  console.log('Dispute entered');
  updateOrder = factory.newTransaction(NS, 'Dispute');
  updateOrder.financeCo = factory.newRelationship(NS, 'FinanceCo', financeCoID);
  updateOrder.buyer = factory.newRelationship(NS, 'Buyer', order.buyer.$identifier);
  updateOrder.seller = factory.newRelationship(NS, 'Seller', order.seller.$identifier);
  updateOrder.dispute = req.body.reason;
  break;
case 'Authorize Payment':
  console.log('Authorize Payment entered');
  updateOrder = factory.newTransaction(NS, 'AuthorizePayment');
  updateOrder.buyer = factory.newRelationship(NS, 'Buyer', order.buyer.$identifier);
  updateOrder.financeCo = factory.newRelationship(NS, 'FinanceCo', financeCoID);
  break;
case 'Cancel':
  console.log('Cancel entered');
  updateOrder = factory.newTransaction(NS, 'OrderCancel');
  updateOrder.buyer = factory.newRelationship(NS, 'Buyer', order.buyer.$identifier);
  updateOrder.seller = factory.newRelationship(NS, 'Seller', order.seller.$identifier);
  break;
```

Seller capability:

- Order From Supplier
- Request Payment
- Resolve (a dispute)
- Refund (an order)

Note: Resolve has already been written for the buyer. We don't have to do anything to reuse it. ...Why?



Zero To Blockchain Chapter 10: Seller View

2nd Life Systems, Inc

View as seller: 2nd Life Systems, Inc

Order #	Status	Total	Buyer: Born On The Cloud, Inc	
000	Order Purchased: 2017-09-25T12:03:33.024Z	\$39800.00	<div>No Action</div> <div>PC Hardware Now, Inc</div> <div>Execute</div>	
Item Number	Description	Quantity	Price	
3	Lenovo Thinkpad W520 16Gb, .25Tb	15	\$7500.00	
4	Lenovo Thinkpad W520 32Gb, 1Tb	20	\$31300.00	

Order #	Status	Total	Buyer: Kid Friendly Learning, Inc	
abbykidfriendlyinccons1506340965954	Order Purchased: 2017-09-25T12:03:06.267Z	\$4265.00	<div>No Action</div> <div>PC Hardware Now, Inc</div> <div>Execute</div>	
Item Number	Description	Quantity	Price	
1	Macbook Pro 16Gb, 1Tb	2	\$2570.00	
0	4K Monitor - 42	0	\$1695.00	

connected

Clear Window

Display Order Status

The Plan: 30 minute Chapters with an hour or two of practice

Chapter 1:	What is Blockchain? Concept and Architecture overview
Chapter 2:	What's the story we're going to build
Chapter 2.1:	Architecture for the Story
Chapter 3:	Set up local HyperLedger V1 development environment
Chapter 4:	Build and test the network
Chapter 5:	Administration User Experience
Chapter 6:	Buyer Support and User Experience
Chapter 7:	Seller Support and User Experience
Chapter 8:	Shipper Support and User Experience
Chapter 9:	Provider Support and User Experience
Chapter 10:	Finance Company Support and User Experience
Chapter 11:	Combining for Demonstration
Chapter 12:	Events and Automating for Demonstration



Chapter 8: Shipper Support and User Experience

Learning Bluemix & Blockchain

Bob Dill, IBM Distinguished Engineer, CTO Global Technical Sales
David Smits, Senior Certified Architect, IBM Blockchain

