BLCN532 Lab 2

Writing chaincode

V2.2

# Introduction

Now that you have a running business network, you can write, install, and run smart contracts, called chaincode, to interacts with your blockchain. Chancode makes up your blockchain application and provides the only way for users to read from, and append to, the blockchain.

In this lab you’ll examine several chaincode files, run the sample chaincode presented in your textbook, and run tests on the chaincode to assess its functionality.

# Lab Deliverables:

To complete this lab, you must create a **Lab Report file** and submit the file in iLearn. The Lab Report file must be a Microsoft Word format (.docx), and have the filename with the following format:

BLCN532\_SECTION\_STUDENTID\_LASTNAME\_FIRSTNAME\_Lab02.docx

* SECTION is the section number of your current course (2 digits)
* STUDENTID is your student ID number (with leading zeros)
* LASTNAME is your last name
* FIRSTNAME is your first name

To get started, create a Microsoft Word document (.docx) with the correct filename for this lab. You’ll be asked to enter text and paste screenshots into the lab report file.

**NOTE: All screenshots MUST be readable. Use the Ubuntu Screen Capture utility (see the lab video.) Make sure that you label each screenshot (i.e. Step 2.1.3) and provide screenshots in order. For commands that produce lots of output, I only want to see the last full screen when the command finishes. Provide FULL screenshots, NOT cropped images.**

# SECTION 1: Running chaincode

## Step 1.1: Launch the sample trade network in development mode

First, you’ll start your sample business network in development mode (note the ‘-d’ option).

1. Open PowerShell
2. PS %HOME%> **cd vagrant\hyperledger**
3. PS %HOME%\vagrant\Hyperledger> **vagrant up**
4. Login to your Linux VM (userid: vagrant password: vagrant) and open a Terminal window.
5. $ **cd $GOPATH/src/trade-finance-logistics/network**
6. $ **./trade.sh up -d true**
7. $ **date**
8. Create a screenshot of the results of steps 5 - 7 and paste it into your Lab Report File.

## Step 1.2: Compile the chaincode

Next, you’ll use a different PuTTY connection to compile and run the chaincode.

1. Open a new Terminal window.

NOTE: This is the SECOND Terminal CONNECTION for screenshots in later steps.

1. $ **cd $GOPATH/src/trade-finance-logistics/network**
2. $ **docker exec -it chaincode bash**

NOTE: If you get an error in step 3, go to your other Terminal app, shutdown the trade network, then re-start it.

1. # **cd trade\_workflow\_v1** (Note the new prompt)
2. # **go build**

## Step 1.3: Run the chaincode

Now you can run the chaincode you just compiled.

1. # **export CORE\_PEER\_ADDRESS=peer:7052**
2. # **export CORE\_CHAINCODE\_ID\_NAME=tw:0**
3. # **./trade\_workflow\_v1** (This is a server process, so you won’t get a prompt back.)

## Step 1.4: Install and run the chaincode on the channel

1. Open a New (3rd) Terminal window.
2. $ **cd $GOPATH/src/trade-finance-logistics/network**
3. $ **docker exec -it cli bash**
4. # **peer chaincode install -p chaincodedev/chaincode/trade\_workflow\_v1 -n tw -v 0**
5. Create a screenshot (in the current Terminal window) of the results of step 4 and paste it into your Lab Report File.
6. # **peer chaincode instantiate -n tw -v 0 -c '{"Args": ["init","LumberInc","LumberBank","100000","Wooden Toys","ToyBank","200000","Universal Freight","ForestryDepartment"]}' -C tradechannel**
7. # **peer chaincode invoke -n tw -c '{"Args": ["requestTrade",”trade-12”,"50000","Wood for Toys"]}' -C tradechannel**

NOTE: the command in step 7 doesn’t copy-paste well. You’ll just have to type it.

1. Create a screenshot (in the current Terminal window) of the results of step 7 and paste it into your Lab Report File.
2. # **peer chaincode invoke -n tw -c '{"Args": ["getTradeStatus","trade-12"]}' -C tradechannel**
3. Create a screenshot (in the current Terminal window) of the results of step 9 and paste it into your Lab Report File.
4. In the SECOND Terminal window (the one you opened in step 1.2), create a screenshot of the results of step 9 and paste that screenshot into your Lab Report File.

# SECTION 2: Reviewing chaincode source code

## Step 2.1:

1. Open a New Terminal window.
2. $ **cd $GOPATH/src/trade-finance-logistics/chaincode/src/github.com/trade\_workflow\_v1**
3. $ **gedit tradeWorkflow.go**
4. Follow the source code development process in the textbook, starting with the “Creating a chaincode” section.
5. Find the **Invoke()** method in the editor. (Make sure the line starts with “func”.)
6. Create a screenshot (in the current Terminal window) of the results of step 5 and paste it into your Lab Report File.
7. Find the **requestTrade()** method in the editor. (Make sure the line starts with “func”.)
8. Create a screenshot (in the current Terminal window) of the results of step 7 and paste it into your Lab Report File.
9. Find the **getTradeStatus()** method in the editor. (Make sure the line starts with “func”.) You invoked this method in the previous section (Step 1.4.9).
10. Scroll down to the end of the getTradeStatus() method. You should see the string “Query Response:” in the next-to-last line in the method. This is the code that prints the trade status.
11. Create a screenshot (in the current Terminal window) of the results of step 10 and paste it into your Lab Report File.
12. Exit gedit

# Section 3: Testing chaincode

## Step 3.1: Test the chaincode

1. $ **cd $GOPATH/src/trade-finance-logistics/chaincode/src/github.com/trade\_workflow\_v1**
2. $ **gedit tradeWorkflow\_test.go**
3. Find the method that tests ShipmentInitiation.
4. Create a screenshot (in the current Terminal window) that shows the other methods (should be 7) the current method invokes to test its functionality. (You can find this information in a program comment. Comments start with two forward slashes. Look for lines that start with “//”.) Paste that screenshot into your Lab Report File.
5. Exit gedit
6. $ **go test**
7. $ **date**
8. Create a screenshot (in the current Terminal window) of the results of steps 6 and 7 and paste that screenshot into your Lab Report File.

# Section 4: Wrapping up

Now that you have compiled, installed, run, and tested your chaincode, you need to properly shutdown your business network.

1. Press **CTRL-C** in the SECOND Terminal window to return to the # prompt.
2. Type exit at the # prompt (in all Terminal windows except the first one) to exit docker.
3. $ **./trade.sh down -d true**
4. Exit from your Linux sessions
5. $ **exit**
6. In Windows PowerShell, shut down your Hyperledger virtual machine:
7. PS %HOME%\vagrant\Hyperledger> **vagrant halt**
8. Create a **snapshot** **(NOT A SCREENSHOT)** of your Ubuntu VM in VirtualBox. This allows you to easily restore to this point if you ever need to “undo” subsequent steps and return to a known point.
	1. PS %HOME%\vagrant\Hyperledger> **vagrant snapshot save EndOfLab02**

You should have 10 screenshots in your Lab Report File. Save your file and submit it in iLearn as a file attachment for the Lab 2 assignment.

Congratulations! You have complete lab 2.