First install the [Hyperledger composer](https://hyperledger.github.io/composer/installing/installing-prereqs.html). Then install the [development environment](https://hyperledger.github.io/composer/installing/development-tools.html).

There’s no need to start Playground while you’re installing the environment.

Make sure docker is running, and when you run **./startFabric.sh** it’s going to take a couple of minutes. So be patient.

Now that your machine is all set, we can start coding!

**Step 1: Create Business Network**

Business Network Definition (BND) consists of the data model, transaction logic, and access control rules. The data model and access control rules are coded in domain specific language (which is very simple to catch up with). The transaction logic will be coded in JavaScript.

To create a BND, we need to create a suitable project structure on disk. We will create a skeleton business network using **Yeoman**. To create a project structure, open your terminal and run the following command:

$ yo hyperledger-composer

This will shoot out a series of questions as follows. You’ll be required to use your arrow keys to navigate through the answers.

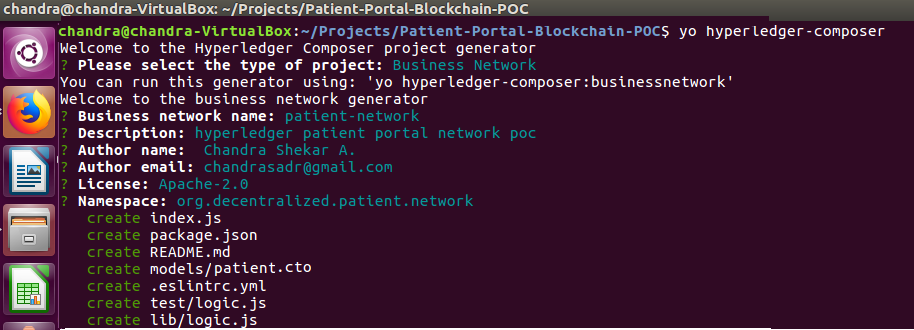
#### Step 1: Outline your Business Network

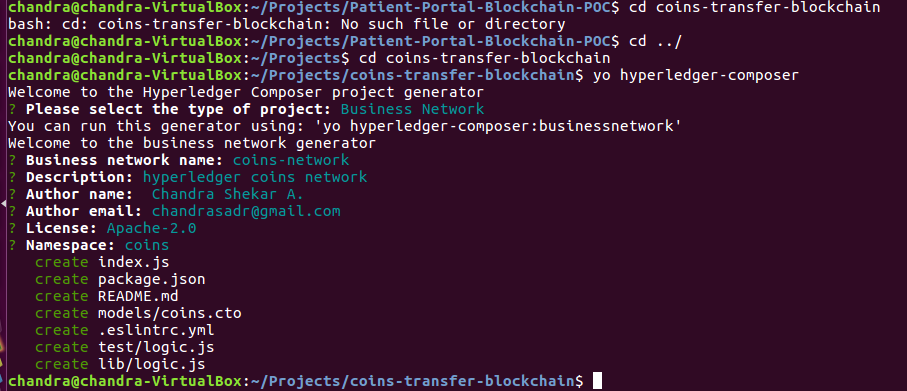
Our Business Network Definition (BND) consists of the data model, transaction logic, and access control rules. The data model and access control rules are coded in domain specific language (which is very simple to catch up with). The transaction logic will be coded in JavaScript.

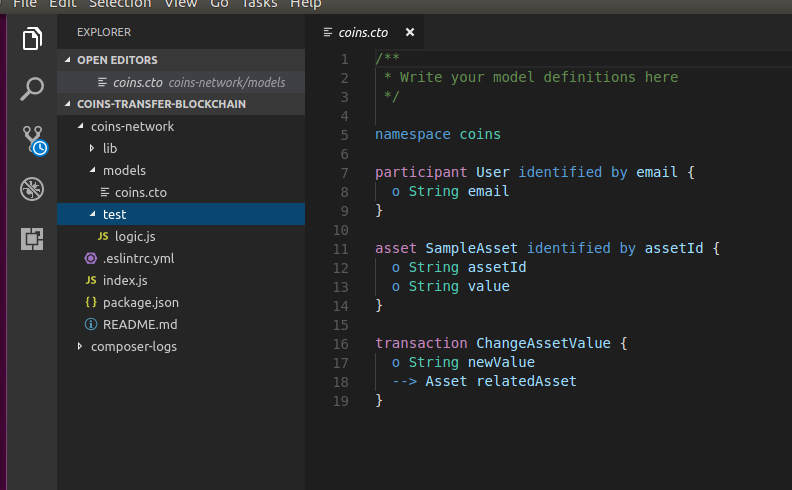
To create a BND, we need to create a suitable project structure on disk. We will create a skeleton business network using **Yeoman**. To create a project structure, open your terminal and run the following command:

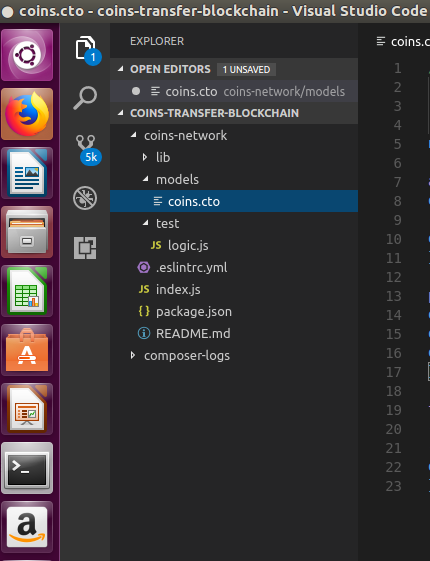
$ yo hyperledger-composer

This will shoot out a series of questions as follows. You’ll be required to use your arrow keys to navigate through the answers.

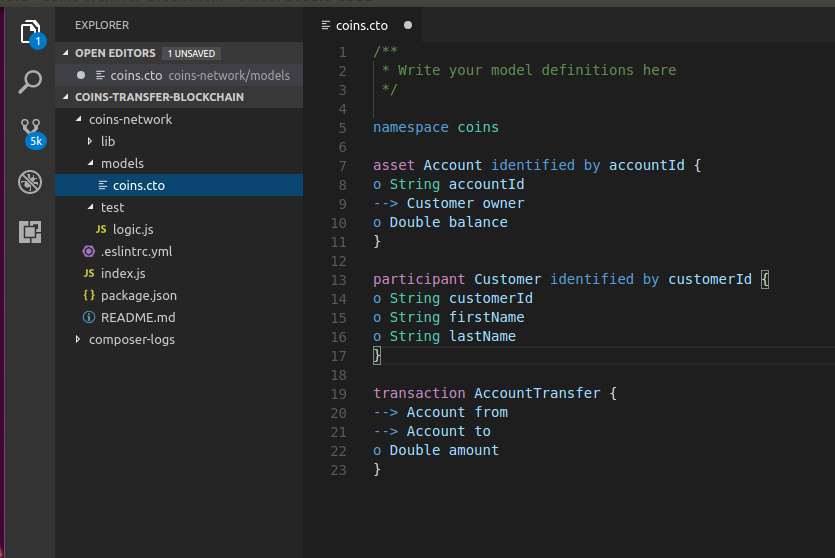








#### Step 2.1: Coding out Business Network (models/coins.cto)



#### ****Step 2.2: Coding out the Business Network (lib/logic.js)****

In this file, we’ll add transaction logic in JavaScript.

#### Step 3: Generate the Business Network Archive (BNA)

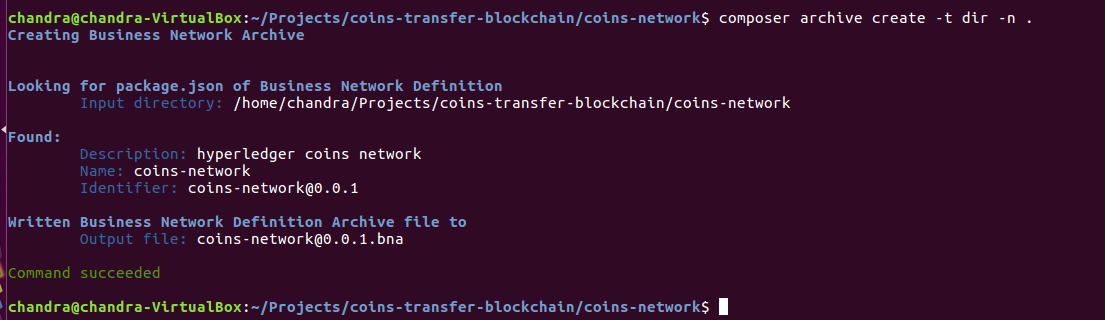
Now that the business network has been defined, it must be packaged into a deployable business network archive (.bna) file.

Step 3.1: Navigate into the test-bank app in your terminal.

Step 3.2: Run the following command:

$ composer archive create -t dir -n .

This creates a .bna file in the **coins-network** folder.

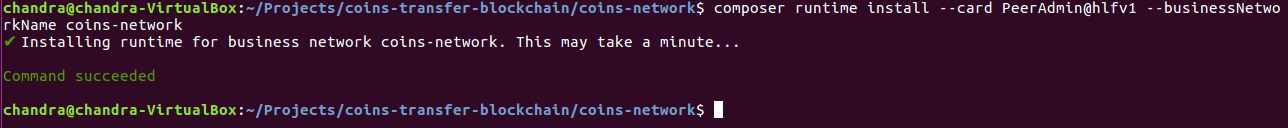


#### Step 4: Deploy the Business Network Archive file on the Fabric

**Step 4.1: Install composer runtime**

#### Step 4: Deploy the Business Network Archive file on the Fabric

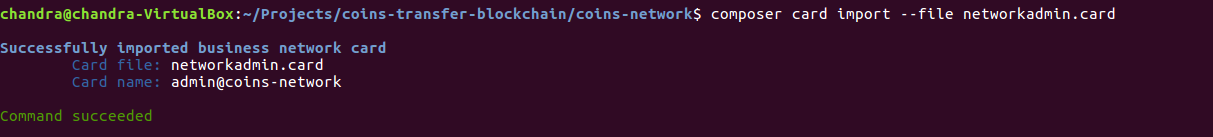
**Step 4.1: Install composer runtime**



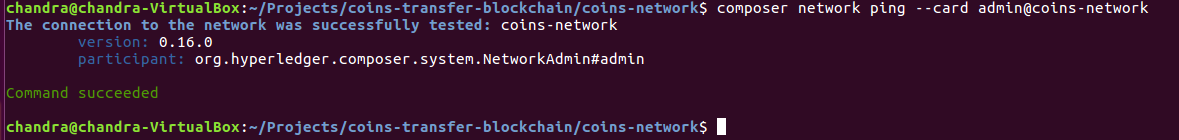
**Step 4.2: Deploy the business network**



**Step 4.3: Import the network administrator identity as a usable business network card**



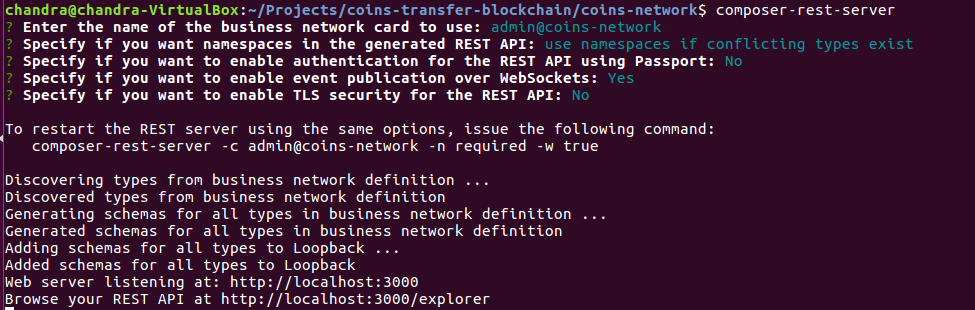
**Step 4.4:**To check that the business network has been deployed successfully,**run the following command to ping the network:**

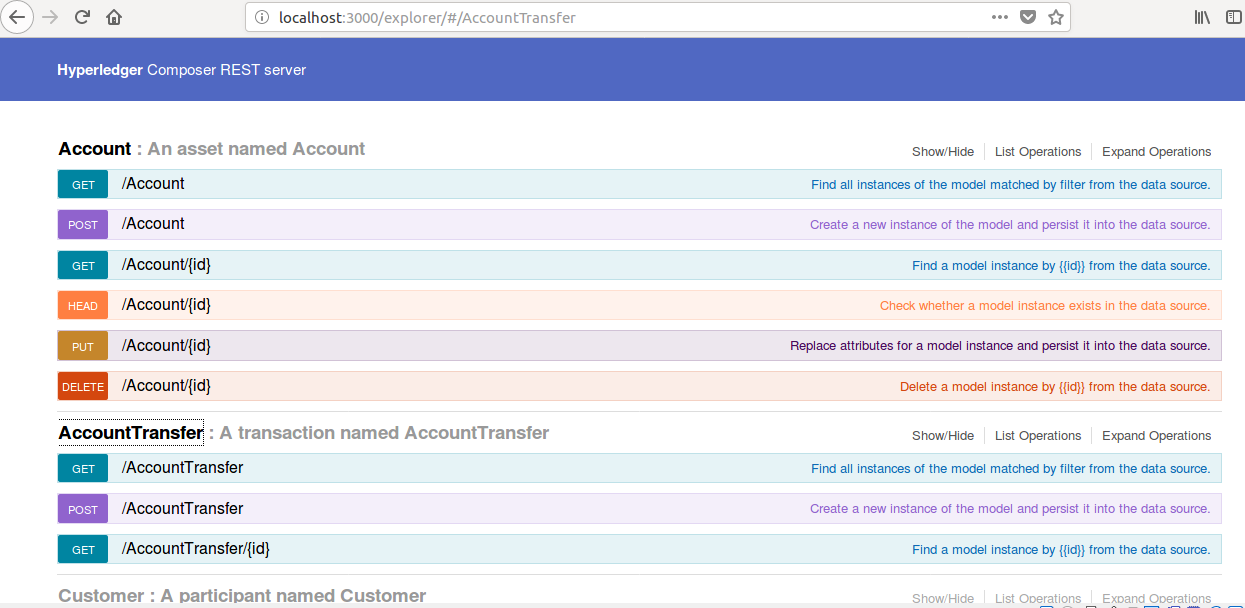


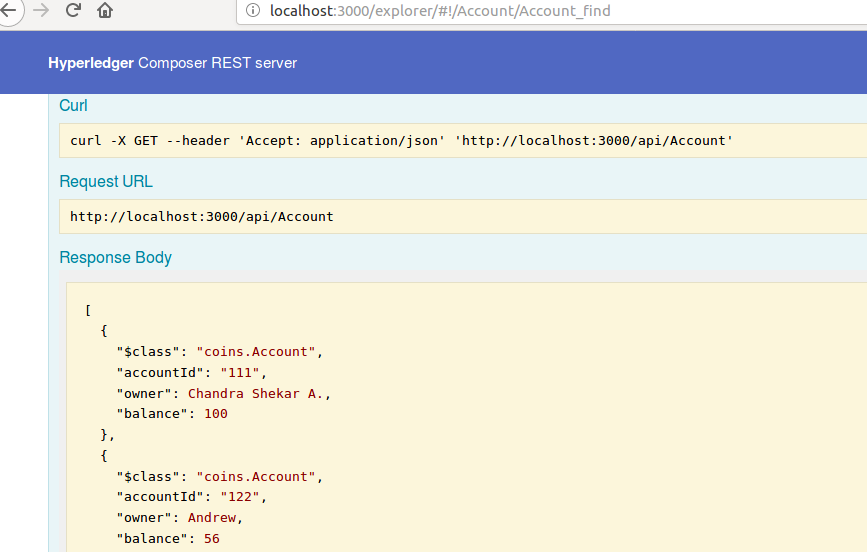
#### STEP 5: Expose a RESTful API

To create a RESTful API from your command line, run the following command:

$ composer-rest-server







#### Step 6: Angular Front End

To create Angular scaffolding automatically, run the following command in the coins-network folder:

$ yo

