ZeroToBlockchain Chapter 5: Building the Admin interface

Return to Table of Contents

In this chapter, we will build a nodeJS web server which will interact with the HyperLedger Composer defined Blockchain and create an administrative interface for the network. This runs on HyperLedger Fabric Version 1.0. The code that we will write is managed in this chapter and all the subsequent chapters in the following folder structure:

```
Chapter 05
  ↳ controller
     ↳restapi
       router.js
        ↳features
           ↳composer
             autoLoad.js
             hlcAdmin.js
             queryBlockChain.js
             Z2B_Services.js
             Z2B_Utilities.js
           ↳ text
               multi-lingual.js
               resources.js
  ↳ HTML
     ↳CSS
       pageStyles.css
     ↳js
       z2b-admin.js
       z2b-events.js
       z2b-initiate.js
       z2b-utilities.js
  ↳ network
```

```
    permissions.acl
    queries.qry
     ↳lib
       sample.js
     ↳models
       base.cto
       events.cto
       sample.cto
```

Some of these files are used by all chapters and some are specific to this chapter. Code explanations are embedded in the javascript files. Those files are:

Common Web Server code:

- **index.js**
  - this is the master, or controller, file for the web server. It loads a file called 'router.js' to implement all business functionality.
- **autoload.js**
  - sometimes it helps to have members and assets preloaded into the business network. This module loads that data from a file called memberList.json in the /startup folder
- **queryBlockchain.js**
  - As you are running the autoloader exec and working with chapters 5-12, you will see blocks being created. This modules listens to the blockchain events and sends that data to the browser.
- **Z2B_Services.js**
  - This module contains transaction management routines which are called either by the autoLoader.js module or hlcClient.js module.
- **Z2B_Utilities.js**
  - This module contains utility services, including socket

    - creation, which are called by autoLoader, hlcClient, hlcAdmin, queryBlockChain modules
- **multi-lingual.js**
    - This module contains services to support multiple languages on the browser. Language text is provided in subfolders in the composer/text folder. Which languages are available is managed in the languages.json file.

Common Web Browser code:

- **z2b-utilities.js**
    - This file contains routines which are used by the browser javascript.
- **z2b-initiate.js**
    - This file contains the code necessary to start up the web application.
    - This is where the default starting language is specified

Files used in this chapter

Web Server Code Unique to this Chapter

- **hlcAdmin.js**
    - This contains all code to support the administrative interface on the server side

Defining the business network

- **network/model/sample.cto**
    - The CTO file contains the network description and includes member (buyer, seller, provider, shipper, financeCo), asset (Order) and transaction definitions. The sample.cto file uses two other files as it's foundation: **base.cto** and **events.cto**
- **network/lib/sample.js**
    - This contains the transaction code which will be deployed to

    the blockchain. Although normally written in GO, hyperledger composer allows us to write all of the blockchain code in javascript and then uses a plug-in called DukTape to enable the javascript to be executed within the GO environment. This file must implement transactions as defined in the sample.cto file and is one of the places where we ensure that members have the authority to execute the requested transaction.

- **network/permissions.acl**
    - This file determines precisely what transactions a member can execute and what assets a member can see. The basic rule is "If authority is not specifically granted in this file, then access is denied."

Web Browser Code

- **z2b-admin.js**
    - This file contains all of the browser functional code to support the administrative interface
- **index.html**
    - This file is the initial web page loaded by the application.
    - Text on this page is determined by the selected language (default is US English)
- **admin.html**
    - Contains the HTML to manage the admin interface
    - Text on this page is determined by the selected language
- **ceateConnectionProfile.html**
    - Contains the HTML to manage profile creation
    - Text on this page is determined by the selected language
- **createMember.html**
    - contains the HTML to manage member creation
    - Text on this page is determined by the selected language
- **deleteConnectionProfile.html**
    - contains the HTML to manage connection deletion

    - Text on this page is determined by the selected language
- **getMemberSecret.html**
    - contains the HTML to manage and display member secrets
    - Text on this page is determined by the selected language
- **removeMember.html**
    - contains the HTML to remove a member
    - Text on this page is determined by the selected language