

# Registration Station

## Introduction

The creators of this project are: Tre Allison, Rebecca Fenter, and Seymone Gugneja. We are three students that attend Wake Tech Community College in Raleigh, North Carolina and we are currently participating in a capstone course that WTCC is offering alongside IBM. Distinguished IBM engineer, Bob Dill, created the [ZeroToBlockchain](#) course that focuses on creating a blockchain for the transactions between a Buyer, Seller, Provider, Shipper and Finance Company. Our Capstone project was based on the [ZeroToBlockchain](#) course, except instead of employing the transactions that Bob defined in his application, we were required to create a course-registration blockchain application that implements the various transactions to take place between a student, registrar and cashier. We named our project **Registration Station**. The following paragraphs will inform you more in depth of our project.

For our project, we were told to create two students and define three participants. The participants are the Student, Registrar, and Cashier. We were told to assume that the student would not encounter any scheduling errors, and can pay the tuition in full with no help from the Financial Aid office. Each student is identified by their name and email and each course is identified by a course code, course title, course schedule, and credit hours. We created registration-station over two OS's: Mac OS and Ubuntu and wrote the app in Node.js, HTML/CSS and JavaScript. We deployed our application to two places: IBM Bluemix cloud and localhost.

## Setting up environment to run locally

To run registration-station on the cloud, navigate to <http://registrationstation.mybluemix.net>. To run registration-station on localhost, make sure you have a Mac OS or Linux system. In the command line, start by installing Docker. To do this, follow these commands. Start by updating the apt package index: `sudo apt-get update`. Now install the necessary packages. Make sure to press enter after the backslashes: `sudo apt-get install \ apt-transport-https \ ca-certificates \ curl \ software-properties-common`. Now you will add Dockers official GPG key: `curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`. Next, you will verify that you have the key by using the fingerprint: `sudo apt-key fingerprint 0EBFCD88`. You will now set up a stable repository by typing the following command and pressing enter after every backslash: `sudo add-apt-repository \ deb [arch=amd64] https://download.docker.com/linux/ubuntu \ $(lsb_release -cs) \ stable`. Next update the apt package index: `sudo apt-get update`. Now install the latest version of Docker CE: `sudo apt-get install docker-ce`. You will now verify that Docker CE is installed correctly by running hello-world: `sudo docker run hello-world`. After all these commands are successfully compiled, install curl: `sudo apt-get install curl`. Now install all the required software - make sure you type this as a single line: `curl -H 'Accept: application/vnd.github.v3.raw' https://raw.githubusercontent.com/rddillIBM/ZeroToBlockchain/master/setup_Ubuntu_Part_1.sh | bash`. Now reboot Ubuntu: `sudo reboot`. After Ubuntu restarts, type this as a single line: `curl`

```
-H 'Accept: application/vnd.github.v3.raw'
```

```
https://raw.githubusercontent.com/rddillIBM/ZeroToBlockchain/master/setup_Ubuntu_Part_2.sh | bash.
```

After all of these commands have compiled successfully, you will get the blockchain application.

Login to [GitHub](https://github.com) and navigate to <https://github.com/Z2BTeamTwo/registration-station.git>. Click on the Fork button to get your own version of the application. Then click on the **Clone** or **Download** button. Copy the url. In the command line, navigate to the directory where you want to place the application and type this command `git clone`

`https://github.com/Z2BTeamTwo/registration-station.git`. Now, in the command line, go to the directory you copied the application into and add this command `cd registration-station`. There are several branches in the project master and local both contain the local version of the project and cloud and development contain the cloud version. Switch to local or master with `git checkout <branchname>`, with the name of the branch that you want to use replacing `<branchname>`. You will now install the node packages used in the project by entering `npm install`. After it has been installed successfully, you will build and deploy the network: `./buildAndDeploy`. After that has been compiled successfully, run this command `npm start`. Navigate to `localhost:6001`.

## Using the App

Now that you are in either localhost or cloud, it's time to explain how to use the application. On the top bar of the webpage, there is a **Roles** drop down. When you click that drop down, you see four options; **Load Unified User Experience**, **Load Student User Experience**, **Load Registrar User Experience**, and **Load Cashier User Experience**. If you want to go through the process without clicking between the student, cashier and registrar pages, you can use the unified user experience to see all the roles on one page. For our project, we were told to assume that the student can pay the tuition without the help of financial aid or scholarships. To start the process, click on the **Admin** drop down and then click **Preload network**. Four items should appear on the screen, and on the bottom of the screen, some blue blocks should appear. Then click on the link to the student user experience page or go to the student section in the unified user experience under the **Roles** drop-down list. Pick a student from the list above the student section and then click **Add Course** at the bottom of the section. A drop-down list then appears in the section. Choose the course you want to add to the registration and after reviewing the details that appear, click either **Submit** or **Cancel**. Repeat this process as many times as you want. After you have registered for the courses you want, click on **Display Courses** at the bottom of the section. In the course info, you should see two options in the list that appears next to the course information: **Register** or **Drop**. If you drop, the course stays in your course list but no further action will be taken on it after the registrar accepts and/or forwards the registration status. If you want to register for the courses, click the **Register** button and then **Execute** next to course drop-down. Then, click **Display Courses**. The registration is then forwarded to the registrar. The registrar can then either accept or decline the registration status update. If the registration is declined, the registrar provides a reason and then sends it back to the student. The student then fixes the problem and re-registers or leaves the course in dropped status. If the registration is accepted, the registrar has two options: the first option is canceling the course. The student is notified and no further action can be taken. The second option is to forward the

course to the cashier, and the cashier can then request the tuition from the student. The student can either pay in full or incrementally. If the registrar cancels the course or the student drops the course after payment, the cashier is notified and the student gets a refund.