User/Developer Guide

Cal Poly Computer Engineering Capstone

Cloud1

Archit Mendiratta, Maximillian Parelius, Etienne Urbain-Racine, Tam Nguyen, Corey Wilson

3/11/2015

Table of Contents

System Overview

Cloud1 allows registered users to control devices through various actions, including device-to-device communication. Users can access the website, as shown in Figure 1, at http://ec2-54-153-0-23.us-west-1.compute.amazonaws.com. The web address is subject to change. After registering for the first-time users (Figure 2) and logging in for returning users,the main device page will be displayed, as shown in Figure 4. If a user wishes to reset their password, he or she can also fill out the reset password form, as shown in Figure 3.
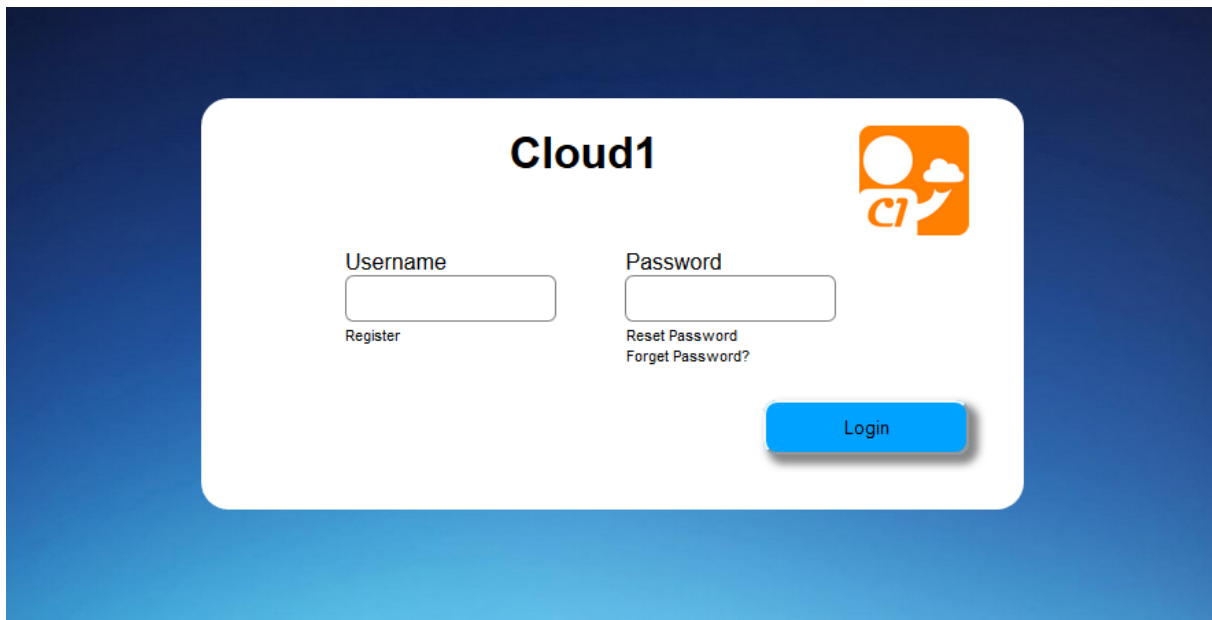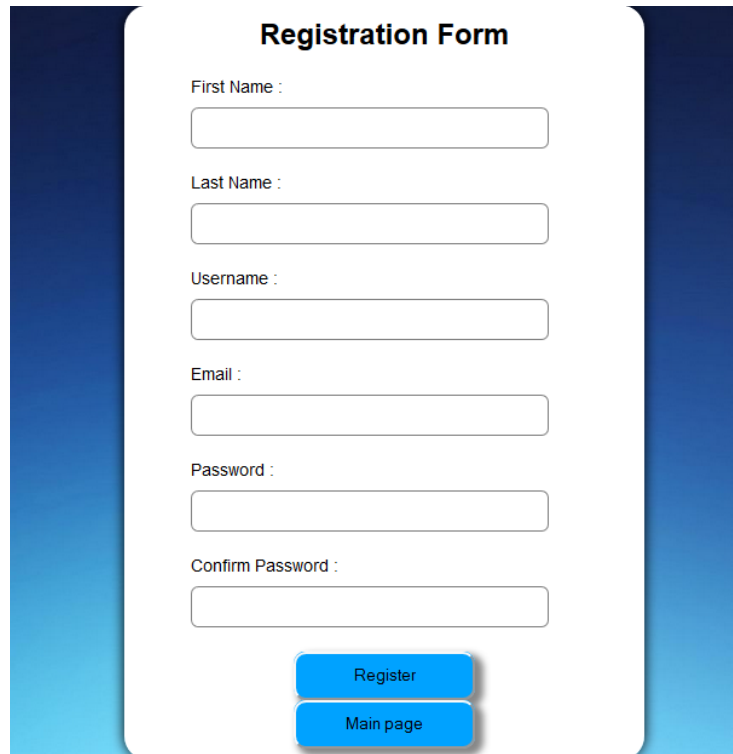

Figure 1. Login page

Figure 2. Registration page

Figure 3. Password reset page
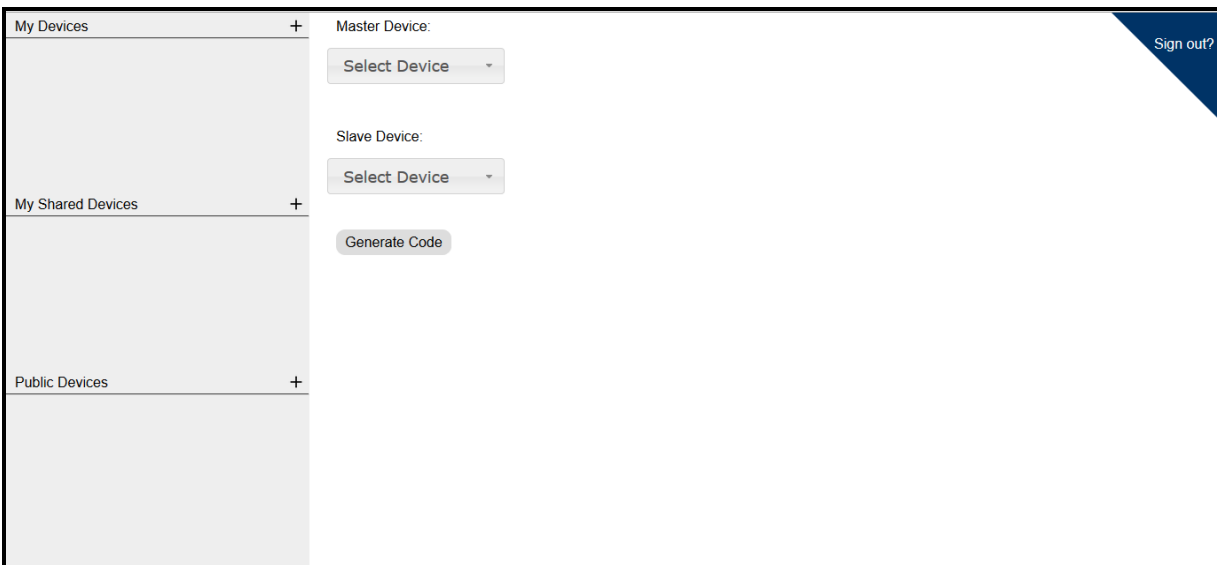
Figure 4. Main page

System Installation

Setting up OpenIoT
- Create an ubuntu EC2 instance through amazon web server or set up your own server
  - we recommend 2GB of RAM and 30GB of Storage.
    - when using AWS do the following.
      1. log into your aws account
      2. got to the EC2 link from your home page
      3. on the left hand side of the screen select instances
      4. click on launch instance
      5. select ubuntu as the AMI
      6. select T2.small as the type of instance
      7. continue until the storage and select 30 GB
      8. Tag the instance with the name you wish to identify your server
      9. for the security group allow SSH and HTTP
      10. Review and launch the instance create a key and keep it. if lost you will not be able to connect to your instance.
      11. save the instance public DNS you will need it for the next step
  - Connect to your server by using PuTTY or another ssh software
    - the address to connect is ubuntu@'Instance Public DNS'
      1. If you are logging in for windows you will need to convert your .pem key with a .ppk key using a program like puttygen

a. click LOAD and select your PEM key
b. make sure the key is SSH-2 RSA and save public key
2. if using PuTTY expand the SSH tab on the right side of the browser and go into Auth from there you can select your key.
3. since you will be using SSH fairly often it's recommend you save the session.
4. click open to connect to your ubuntu instance.
- ○ Open IoT requires 1GB of the RAM of your server as swap memory to do this enter in the following commands once connected through ssh
  - ■ `sudo /bin/dd if=/dev/zero of=/var/swap.1 bs=1M count=1024`
  - ■ `sudo /sbin/mkswap /var/swap.1`
  - ■ `sudo /sbin/swapon /var/swap.1`
- Once connected to your server, follow the steps in this manual to set up OpenIoT
  - ○ See appendices for further information about running OpenIoT on an AWS

Setting up Web Server
- For database and web hosting capabilities install the LAMP package
  - ○ LAMP (Linux-Apache-MySQL-PHP) is installed with the following command
    - ■ `sudo apt-get update`
    - ■ `sudo apt-get install lamp-server^`

Setting up Hardware - Arduino
- Necessary tools
  - ○ Arduino board
  - ○ cc3000 wifi shield
  - ○ Grove sensor mount
  - ○ grove sensors
- Assemble the parts
  - ○ Mount the wifi shield on top of the arduino
  - ○ Mount the grove sensor board on top of the wifi shield
  - ○ Connect any number of grove sensors to the sensor board
- Follow the steps here to connect the arduino to a PC
- Running the REST server on the arduino
  - ○ Install the packages here and save them in your arduino folder on your PC
  - ○ Open up the arduino sketch application on your PC
  - ○ open the rest_server.ino file inside the REST folder on sketch
  - ○ Make sure the arduino is connected to the PC
  - ○ Modify the rest_server.ino file so it connects to your desired wifi
    - ■ Enter ssid, password, and security type

```
#define WLAN_SSID       ""           // cannot be longer than 32 characters!
#define WLAN_PASS       ""
// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2
#define WLAN_SECURITY   WLAN_SEC_UNSEC
```

- ○ Modify the rest_server.ino file to expose the necessary sensors to rest
  - ■ Follow the examples provided in the file
- ○ Next specify the port that the arduino is connected to by going to tools->Serial Port->COMX
  - ■ To find out what port your arduino is connected to on your PC for windows go to Control Pane->Device Manager->Ports
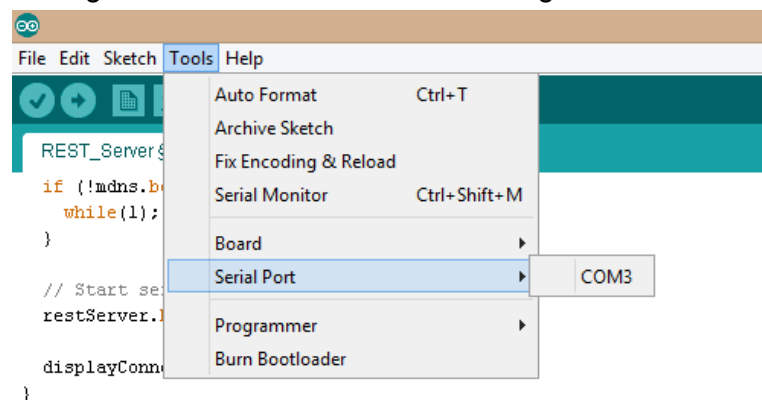
- ○ Once the file is configured correctly, press the right arrow key at the top left of the arduino sketch application to load the package onto the arduino
- ○ Once the package has been successfully loaded you can log onto a serial console using Putty to see which IP your REST server is running on
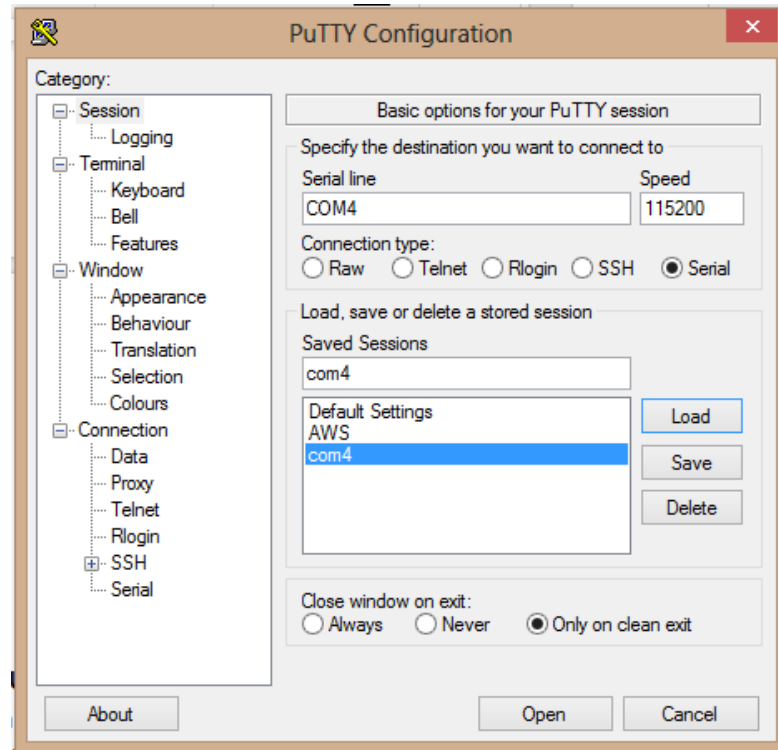
- ○ Once you have the IP endpoint of the rest client you can type the IP into a browser from any network connection to get the sensor data from the arduino

- ○ To access any one particular sensor from the REST server simply type <IP endpoint>/<sensor name> into the browser
  - ■ If I wanted light value of this example, I would type 172.16.0.21/light into the browser to get the light data

System Functionality

- - When the user accesses the Cloud1 homepage, they will see a login page requiring a username and password.
- - The user will click the register link to be taken to a web page that prompts a user for information to setup an account.
- - If the user ever wishes to reset their password, they can use the reset password link to be taken to a web page that requires account information to reset the password.
- - When a user successfully logs in on the homepage, they will be taken to the web app.

- On the web app page, the user's registered devices will appear in the left column. To register a device, the user needs to click on the "+" icon. Upon clicking on the "+" icon, a dialogue will pop up, prompting the user for device information.
- After devices have been registered, the devices can be selected in the dropdown menus to the right of the device column.
- Once the devices and actions have been selected, the user will click on the ""Make it happen!" button to perform the action.
- When the user is done with their session, they need to click the sign out link in the upper-right corner to log out.

Documentation

Web App (./WebDevTutorial)
- **index.php** (redirects to 'main.php' if the user is already logged in)
  - `#register:` Redirects to 'register.html' when clicked.
  - `#login:` Sends a request to 'login.js' when clicked.
  - `#pass:` Enter an account password in this text field.
  - `#pass-forget:` Redirects to 'password-forget.html' when clicked.
  - `#pass-reset:` Redirects to 'password-reset.html' when clicked.
  - `#user:` Enter an account username in this text field.
- **main.php** (redirects to 'index.php' if the user is NOT logged in)
  - `#signout:` Redirects to 'logout.php' when clicked.
  - `#button1/#button2/#button3:` Sends a request to 'main.js' when clicked.
  - `#master-inputs:` Contains drop-menus for selecting a master device.
  - `#slave-inputs:` Contains drop-menus for selecting a slave device.
  - `#generate-code:` <not implemented>
  - `#dialog:` Dialog window for device registration
    - `#name:` Enter the device name in this text field.
    - `#device:` Select a device API from the drop-menu.
    - `#ip:` Enter the device IP address in this text field.
    - `#add-device:` Sends a request to 'main.js' when clicked.
- **password-forget.html** (this page for retrieving a lost account password)
  - `#email:` Enter the email address corresponding to the account with the forgotten password.
  - `#home:` Redirects to 'index.php' when clicked.
  - `#passwordForget:` - Sends a request to 'password-forget.js' when clicked.
- **password-reset.html** (this page is for resetting an account password)
  - `#email:` Enter the corresponding email address in this text field.
  - `#oldPassword:` Enter the corresponding old password in this text field.
  - `#newPassword:` Enter the desired new password in this text field.
  - `#cNewPassword:` Re-enter the the new password in this text field.
  - `#passwordReset:` Sends a request to 'password-reset.js' when clicked.
  - `#home:` Redirects to 'index.php' when clicked.

- **register.html**
  - ○

/js/
- **login.js**
- **main.js**
- **password-forget.js**
- **password-reset.js**
- **register.js**

/php/
- **add-device.php**
- **get-devices.php**
- **login.php**
- **logout.php**
- **password-forget.php**
- **password-reset.php**
- **phpinfo.php**
- **register.php**
- **test-email.php**

[session variables]
- `$_SESSION['deviceCount']`
- `$_SESSION['loggedIn']`
- ● `$_SESSION['user']`
- ● `$_SESSION['userDevices']`

Troubleshooting

Potential issues that may arise and fixes for them:
- Registered devices only show up under 'My Devices'
  - **Reason:** Registered devices are currently set to 'private' by default.
  - **Fix:** Implement the backend to register devices as either 'public' or 'private' accordingly.
- Unauthorized login
  - **Reason:** SQL injection is possible.
  - **Fix:** Further development on the backend to handle parameterized queries and/or unexpected username/password values.
- 'Generate Code' button doesn't do anything!
  - **Reason:** It once had a purpose. It was going to be great! Upon a single click, it would have produced the most outstanding of results. There wasn't much at first, but it had something going for it. For a long time it sat there, half-implemented, waiting for the day its destiny would be fulfilled. Sadly, time and neglect has left this button an empty shell. A remnant of high hopes and unfulfilled dreams.
  - **Fix:** Revive it! Remove it! Really up to you.

Appendices

OpenIoT on AWS Ubuntu Instance

- For installing Java on instance follow these steps: [JavaUbuntuInstall](#)
- For installing maven on instance follow these steps: [MavenUbuntuInstall](#)
- For installing JBoss on instance follow these steps: [JbossUbuntuInstall](#)
- Getting OpenIoT code:
    - `sudo apt-get install git`
    - `git clone` https://github.com/OpenIotOrg/openiot.git