

IoTService (“My IoT Platform”)

The “My IoT Platform” is designed and implemented to resolve the IoT devices integration issue.

- Management of logic relationship among selected IoT device
- Communication among selected IoT devices. (eg. machine-machine communication)

“My IoT Platform” is developed based on Springframe MVC.

“My IoT Platform” offers a list of RESTful based APIs.

“My IoT Platform” communicates with Arduino based IoT devices ActiveMq by using MQTT protocol

How to setup the running environment for this application

- Java(JDK) 1.8 (This application is developed based on Java 1.8). The JRE 1.8 is required for running this application.
- This application needs to be deployed on apache Tomcat 8.0 or later
- Apache ActiveMq is used as MQTT message broker, for this application. MQTT message broker is used to exchange message between the IoT Platform and Arduino IoT devices.
- Apache ActiveMq 5.15.3 is preferred to be used to connect to this application.
- How to connect the application to ActiveMq —ActiveMq needs to be connected to this application (to setup the connection to ActiveMq, go

to -- WEB-INF->iotService.properties, for configuration item mqtt.broker.url to set as mqtt.broker.url=<tcp://<ip address of activeMq host and port>

- Mongo DB is used as data storage for this application
-
- Mongo DB version 3.6.3 is preferred to be used to connect to this application
- How to connect the application with MongoDB
- To setup the connection to MangoDB,
 1. go to -- WEB-INF->iotService.properties, set
 2. *mongo.db.host=<the ip address of host name of the host where the mongo db is deployed>*
 3. *mongo.db.port=<the port of the target mongo db server>*
 4. *mongo.db.name=<the target mongo db database schema name>*

How to Generate Deployment file

The source code of “My IoT Platform” can be build by running the Maven (3.0) script.

target war file IoTService.war will be generated once the Maven script is run.

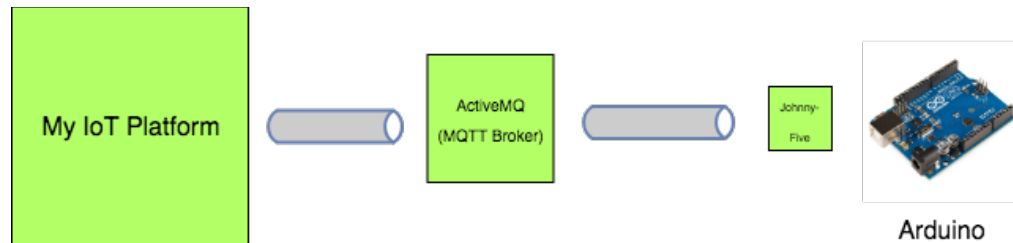
Arduino

Arduino IoT devices are being used as IoT device for testing purpose for this project.

Johnny-Five is a JavaScript library which provides a set of JavaScript API for JavaScript based application to work with Arduino device. In this project we are using the API provided in Johnny-Five library to read data

from Arduino based IoT sensor and send command to the Arduino based IoT reactor.

The end to end test is performed when the environment is setup as shown in figure 3.



To connect Arduino Micro control board

- In this project Arduino UNO R3 board is preferred.
- Arduino Genuino is a IDE platform tool allows user to upload firmware software to Arduino devices.
- In this project Arduino Genuino 1.8.6 is preferred.
- In order to allow Jonny-Five (a node.js library) to communicate with Arduino board. A firmware called “StandardFirmata” is required to be uploaded to the Arduino board via Arduino Genuino. (StandardFirmata comes with Arduino Genuine)
- Once StandardFirmata is uploaded to Arduino board, the Arduino board is able to communicate with node.js based application via Jonny-Five library

Jonny-Five based IoT device sample applications

Code under directory “arduino” is a sample node.js code which uses Jonny-Five library to communicate with Arduino devices.

The node.js code file oneBoard.js is a piece of code used for testing purpose for this project.

This piece of code is developed connect to three Leds and one temperature measurer. The code can also connect to the MQTT message broker, to receive and send command.

Git Hub repository

<https://github.com/bigsoft98/IoTService>