**浙大城市学院实验报告**

课程名称 物联网技术与应用 实验项目 实验八 MQTT规则引擎与数据库操作

专业班级 计算2002 学号 32001019 姓名 司晨旭

指导老师（签名 ） 蔡建平 日期 2022/11/21 实验成绩

**一、实验目的：**

掌握MQTT云平台使用基础知识，规则引擎与数据存储实践。

**二、实验内容：**

1. 搭建试用 EMQX Cloud；

2. 采用MQTTX或其它MQTT客户端连接EMQ Cloud；

3. 采用 EMQX Cloud mysql 插件连接数据库，并进行数据库操作；

**三、实验步骤：**

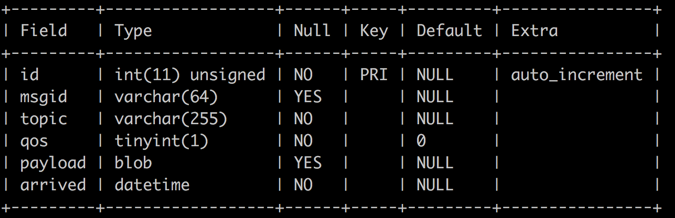
1. 创建数据库，数据库命名规则为“MQTT+姓名首字母”，例·如：MQTTCJP

数据库创建过程截图：

|  |
| --- |
|  |

建数据表，命名规则为“iot+姓名首字母”，例·如：**iot\_cjp**

表结构如下图：



**建表过程截图**

|  |
| --- |
|  |

|  |
| --- |
|  |

2. 在EMQX 的Dashboard中配置连接数据库资源；

|  |
| --- |
|  |

3. 订阅接收以下规则的topic消息：**ZUCC/cjp**（需改为本人姓名缩写）主题消息，如果payload中包含字符串“iot2022”则触发写入MySQL表中的规则：

截图：

|  |
| --- |
|  |

插入数据记录截图：

|  |
| --- |
|  |

查询 MySQL 数据内容截图：

|  |
| --- |
|  |

4. 订阅接收以下规则的topic消息：**ZUCC/#**主题消息，实现多个二级topic消息的订阅接收，如果payload中包含字符串“iot2022”则触发 写入MySQL表中的规则，解析中payload中的字段值写入数据库表中：

截图：

|  |
| --- |
|  |

插入数据记录截图：

|  |
| --- |
|  |

查询 MySQL 数据内容截图：

|  |
| --- |
|  |

5. 配置MQTT鉴权账号，并从任意 topic 的消息中提取 username为自己当前用户信息的字段，写出SQL引擎规则的语句：

|  |
| --- |
| SELECT  id,topic,qos,payload,payload.msg as msg,  Timestamp  FROM  "#"  WHERE  username = 'ZUCCSCX' |

|  |
| --- |
|  |

6. 采集ESP8266的温度、湿度、时间等数据，构建JSON格式的消息，

新建一张数据库表为esp8266，通过规则引擎将数据写入到表中。参考字段：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 学号 | 姓名 | IOT编号 | 湿度 | 湿度 | 时间 | 用户名 |

|  |
| --- |
| #include <ESP8266WiFi.h>  #include <PubSubClient.h>  #include <ArduinoJson.h>  #include "DHT.h"  const char \*ssid = "scxwifi";  const char \*password = "cxcx629326";  const char \*mqtt\_broker = "121.5.180.148";  const char \*topic = "toSCX/1019";  const char \*mqtt\_username = "emqx";  const char \*mqtt\_password = "public";  const int mqtt\_port = 1883;  #define DHTPIN 5  #define DHTTYPE DHT11   // DHT 11  unsigned long previousMillis = 0;  WiFiClient espClient;  // PubSubClient client(espClient);  // DHT dht(5,DHTTYPE);  PubSubClient client(espClient);  DHT dht(5,DHT11);  void callback(char\* topic, byte\* payload, unsigned int length) {    Serial.print("Message arrived [");    Serial.print(topic);    Serial.print("] ");    for (int i = 0; i < length; i++) {      Serial.print((char)payload[i]);    }    Serial.println();    if ((char)payload[0] == '1') {       digitalWrite(BUILTIN\_LED, LOW);     } else {      digitalWrite(BUILTIN\_LED, HIGH);    }  }  void setup() {      Serial.begin(115200);       WiFi.begin(ssid, password);      while (WiFi.status() != WL\_CONNECTED) {       delay(500);       Serial.println("Connecting to WiFi..");     }       Serial.println("Connected to the WiFi network");         client.setServer(mqtt\_broker, mqtt\_port);       client.setCallback(callback);       while (!client.connected()) {       String client\_id = "esp8266-client-";       client\_id += String(WiFi.macAddress());       Serial.println("Connecting to public emqx mqtt broker.....");         if (client.connect("ESP8266Client")) {         Serial.println("Public emqx mqtt broker connected");         } else {         Serial.print("failed with state ");         Serial.print(client.state());         delay(2000);         }   }   dht.begin();  }  void loop() {     client.loop();     unsigned long currentMillis = millis();     if (currentMillis - previousMillis >= 5000) {       previousMillis = currentMillis;     float temp = dht.readTemperature();   float hum = dht.readHumidity();       DynamicJsonDocument data(256);           data["xuehao"] = "32001019";         data["name"] = "SCX";         data["temp"] = temp;         data["hum"] = hum;           char json\_string[256];           serializeJson(data, json\_string);           Serial.println(json\_string);           client.publish(topic, json\_string, false);  }  } |

|  |
| --- |
|  |

1. **本次实验的心得与体会：**

要保护好服务器，否则被攻击了就要重新再来。

有上次的基础之后这次的蛮简单的。