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	实验十 WebSocket 连接 MQTT 服务		
专业班级		学号		姓名	
指导老师（签名 ）	蔡建平	日期		实验成绩	

一、实验目的：

学习 WebSocket 相关内容，掌握 WebSocket 连接 MQTT 服务的客户端页面设计。

二、实验内容：

1.完成一个 HTML 页面，实现 WebSocket 连接 MQTT Broket 服务，实现消息订阅（Subscribe）和发布（Publish）。

（要求连接自己的 MQTT 服务器，要求设置用户名和密码，可使用 EMQ 提供的云服务）

2. 实现订阅消息和发布消息的列表显示。

3. 订阅并解析来自 WebSocket 客户端的 JSON 格式的消息，并将内容写入数据表中;

三、实验步骤：

1. 完成一个 HTML 页面，实现 WebSocket 连接 MQTT Broket 服务，实现订阅（Subscribe），发布（Publish）。

（要求连接自己的 MQTT 服务器，要求设置用户名和密码，可以使用 EMQ 提供的云服务）

代码：

```
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">


  <title>WebSocket</title>
  <!-- mqtt.js -->
  <script src="./WebSocket_files/mqtt.min.js"></script>
</head>
<style type="text/css">
```

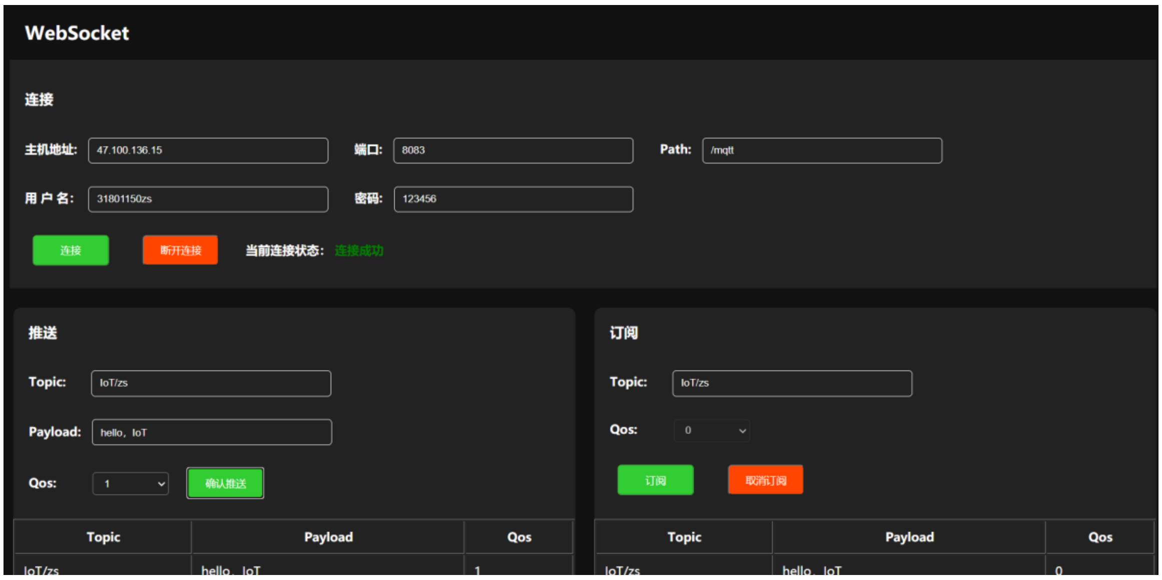
```
body{
	background-color: #111111;
}
#connection{
	background-color: #222222;
	color: white;
	padding: 20px 20px;



}
#connection input{
	background-color:#222222;
	color: white;
	border: 1px #eeeeee solid;
	margin: 20px 30px 10px 10px;



}

#pub input{
	background-color:#222222;
	color: white;
	border: 1px #eeeeee solid;
	margin: 20px 30px 10px 10px;

}

#sub input{
	background-color:#222222;
	color: white;
```

```
        border: 1px #eeeeee solid;
        margin: 20px 30px 10px 10px;

}




#on{
        color: #ffffff;
        background-color: limegreen;
        width: 100px;
        height: 40px;
        padding: 0px;
}
#off{
        background-color: orangered;
        color: #ffffff;
        width: 100px;
        height: 40px;
        padding: 0px;
}

input{
        width: 300px;
        height: 30px;
        border-radius: 5px;
        padding-left: 10px;
}

button{
        margin: 20px 30px 10px 10px;
        border-radius: 5px;

}

select{
        background-color: #222222;
        color: #ffffff;
        width: 100px;
        height: 30px;
        padding-left: 10px;
        margin-left: 10px;
        border-radius: 5px;

}

#pub_button{
        color: #ffffff;
        background-color: limegreen;
        width: 100px;
        height: 40px;
        padding: 0px;
}

#sub_on_button{

        color: #ffffff;
        background-color: limegreen;
        width: 100px;
        height: 40px;
        padding: 0px;
    }

    #sub_off_button{
        color: #ffffff;
        background-color: orangered;
        width: 100px;
        height: 40px;
        padding: 0px;
    }
    #sub_qos{

        margin: 20px 30px 10px 10px;
    }

    th{
        height: 40px;
    }

    td{
        height: 40px;
        padding-left: 10px;
    }



  </style>
  <body>
  <h2  style="color: white;padding-left: 20px;">WebSocket</h2>
  <div id="connection">
          <form id="connection_form" name="connection_form" method="post">
          <h3>连接</h3>
          <b>主机地址: </b><input type="text" value="47.100.136.15" id="host">
          <b>端口: </b><input type="text" value="8083" id="port">
          <b>Path: </b><input type="text" value="/mqtt" id="path">
          <br>
          <b>用 户 名: </b><input type="text" id="username">
          <b>密码: </b><input type="text" id="password">
          <br>
          <button type="button" onclick="connection() " id="on">连接</button>
          <button     type="button"          disabled="disabled"     id="off"
onclick="disconnect()">断开连接</button>

          <b >当前连接状态: </b>
          <b id="info" style="color: blue;">未连接</b>
          </form>
  </div>
  <br>
  <div  id="pub"  style="background-color:#222222;height:100%;width:49%;float:left;color:
white;border-radius: 10px;margin: 5px; ">
      <h3 style="padding-left:20px">推送</h3>
          <form style="padding-left:20px">
          <b>Topic:</b>&nbsp;&nbsp;&nbsp;&nbsp;
          <input type="text" id="pub_topic">
          <br>
          <b>Payload:</b>
          <input type="text" id="pub_payload">
          <br>
```

```
            <b>Qos:</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            <select name="qos" id="pub_qos" style="margin-bottom: 31px;">
                <option value="0" selected="">0</option>
                <option value="1">1</option>
                <option value="2">2</option>
            </select>&nbsp;&nbsp;
            <button    type="button"        id="pub_button"    disabled="disabled"
onclick="pub()">确认推送</button>
        </form>
        <table width="100%" border="1" id="pub_Tbl">
            <tbody>
            <tr>
                    <th scope="col" name="topic">Topic</th>
                    <th scope="col" name="payload">Payload</th>
                    <th scope="col" name="qos">Qos</th>
            </tr>
                </tbody>
        </table>
  </div>
  <div  id="sub"  style="background-color:#222222;height:100%;width:49%;float:left;color:
white;border-right: 1px;border-radius: 10px;margin: 5px 0px 5px 20px;">
        <h3 style="padding-left:20px">订阅</h3>
        <form style="padding-left:20px">
            <b>Topic:</b>&nbsp;&nbsp;&nbsp;&nbsp;
            <input type="text" id="sub_topic">
            <br>
            <b>Qos:</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            <select name="pub_qos" id="sub_qos" >
                <option value="0" selected="">0</option>
                <option value="1">1</option>
                <option value="2">2</option>
            </select>&nbsp;&nbsp;
            <br>
            <button    type="button"      id="sub_on_button"    disabled="disabled"
onclick="sub_on()">订阅</button>
            <button    type="button"      id="sub_off_button"    disabled="disabled"
onclick="sub_off()">取消订阅</button>
        </form>
        <br>
        <table width="100%" border="1" id="sub_Tbl">
            <tbody>
            <tr>
                    <th scope="col">Topic</th>
                    <th scope="col">Payload</th>
                    <th scope="col">Qos</th>
            </tr>
        </tbody>
        </table>
  </div>



<script>
  //连接
  var options;
  var client;
  var connectioning=false;
  function connection() {
        if(host.value ==""||port.value==""||host.value ==" "||port.value==" "){
            window.alert("主机地址和端口不能为空");
            return;
        }
        options = {

            clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),
            username: username.value,
            password: password.value
        }
        console.log("ws://"+host.value+":"+port.value+path.value);

        try{
            client = mqtt.connect("ws://"+host.value+":"+port.value+path.value,options);
            document.getElementById ("on").setAttribute("disabled", "disabled");
            document.getElementById ("info").innerHTML =("连接中");
            document.getElementById ("info").style.color = 'orange';
            connectioning=true;
            setTimeout(function () {
                if(connectioning == true){
                    window.alert("连接超时");
                    document.getElementById ("info").innerHTML =("连接超时");
                    document.getElementById ("info").style.color = 'red';
                    document.getElementById ("on").removeAttribute("disabled");
                    connectioning = false;
                    client.end();
                }
            },5000);
            client.on('connect', function () {
                connectioning=false;

                document.getElementById ("info").innerHTML =("连接成功");
```

```
                document.getElementById ("info").style.color = 'green';

                document.getElementById ("off").removeAttribute("disabled");
                document.getElementById ("pub_button").removeAttribute("disabled");
                document.getElementById
("sub_on_button").removeAttribute("disabled");

                document.getElementById ("host").setAttribute("disabled", "disabled");
                document.getElementById ("port").setAttribute("disabled", "disabled");
                document.getElementById ("path").setAttribute("disabled", "disabled");
                document.getElementById          ("username").setAttribute("disabled",
"disabled");
                document.getElementById          ("password").setAttribute("disabled",
"disabled");
            })
        }
        catch (err) {
            window.alert("连接失败!!!\n 错误信息："+err)

            document.getElementById ("info").innerHTML =("连接失败")
            document.getElementById ("info").style.color = 'red';
        }
    }
    function disconnect(){
        connectioning=false

        document.getElementById ("info").innerHTML =("未连接");
        document.getElementById ("info").style.color = 'blue';

        document.getElementById ("on").removeAttribute("disabled");
        document.getElementById ("off").setAttribute("disabled", "disabled");
        document.getElementById ("pub_button").setAttribute("disabled", "disabled");
        document.getElementById ("sub_on_button").setAttribute("disabled", "disabled");
        document.getElementById ("sub_off_button").setAttribute("disabled", "disabled");

        document.getElementById ("host").removeAttribute("disabled");
        document.getElementById ("port").removeAttribute("disabled");
        document.getElementById ("path").removeAttribute("disabled");
        document.getElementById ("username").removeAttribute("disabled");
        document.getElementById ("password").removeAttribute("disabled");

        client.end(true);
    }
    function pub() {
        if(pub_topic.value == "" ||pub_payload.value == ""||pub_topic.value == " "
||pub_payload.value == " "){
            window.alert("Topic 和 Payload 不能为空");
            return;
        }
        var pub_options={
            qos: parseInt(pub_qos.value)
        };
        try{
            client.publish(pub_topic.value,pub_payload.value,pub_options,function (e){
                var oTr = document.getElementById("pub_Tbl").insertRow(1);
                var aText = new Array();
                aText[0] = document.createTextNode(pub_topic.value);
                aText[1] = document.createTextNode(pub_payload.value);
                aText[2] = document.createTextNode(pub_qos.value);
                for (var index = 0; index < aText.length; index++) {
                    var oTd = oTr.insertCell(index);
                    oTd.appendChild(aText[index]);
                }
            });
        }
        catch (err) {
            window.alert("发送失败\n 错误信息:"+err)
        }
    }
    function sub_on() {
        console.log(client.messageIdToTopic)
        options = {

            clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),
            username: username.value,
            password: password.value
        }
        console.log("ws://"+host.value+":"+port.value+path.value);
        if(sub_topic.value == ""||sub_topic.value == " "){
            window.alert("Topic  不能为空");
            return;
        }
        var sub_option = {
            qos: parseInt(sub_qos.value)
        }
        client.subscribe(sub_topic.value,sub_option,function (err,granted) {
            if (!err) {
                console.log(granted);
                document.getElementById ("sub_qos").setAttribute("disabled", "disabled");
                document.getElementById              ("sub_topic").setAttribute("disabled",
"disabled");
                document.getElementById
("sub_off_button").removeAttribute("disabled");
                document.getElementById      ("sub_on_button").setAttribute("disabled",
"disabled");

            } else {
                window.alert("订阅失败\n 错误信息:" + err)
            }
        })
        client.on('message', function (topic, payload,packet){
            console.log(packet);
            var oTr = document.getElementById("sub_Tbl").insertRow(1);
            var aText = new Array();
            aText[0] = document.createTextNode(topic);
            aText[1] = document.createTextNode(payload);
            aText[2] = document.createTextNode(packet.qos);
            for (var index = 0; index < aText.length; index++) {
                var oTd = oTr.insertCell(index);
                oTd.appendChild(aText[index]);
            }
        })
    }
    function sub_off() {
        client.unsubscribe(sub_topic.value,function (err){
            if(err){
                window.alert("取消订阅失败\n 错误原因:"+err);
            }
            document.getElementById ("sub_qos").removeAttribute("disabled");
            document.getElementById ("sub_topic").removeAttribute("disabled");
                document.getElementById ("sub_on_button").removeAttribute("disabled");
                document.getElementById            ("sub_off_button").setAttribute("disabled",
"disabled");

            })
    }
</script>
</body></html>
```

截图:



WebSocket

连接

主机地址: 47.100.136.15　端口: 8083　Path: /mqtt

用 户 名: 31801150zs　密码: 123456

当前连接状态: 连接成功

推送

Topic: IoT/zs

Payload: hello，IoT

Qos: 1

Topic	Payload	Qos
IoT/zs	hello，IoT	1

订阅

Topic: IoT/zs

Qos: 0

Topic	Payload	Qos
IoT/zs	hello，IoT	0

4. 实现订阅消息和发布消息的列表显示。

代码:

```
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>WebSocket</title>
    <!-- mqtt.js -->
    <script src="./WebSocket_files/mqtt.min.js"></script>
</head>
<style type="text/css">
```

```
body{
	background-color: #111111;
}
#connection{
	background-color: #222222;
	color: white;
	padding: 20px 20px;



}
#connection input{
	background-color:#222222;
	color: white;
	border: 1px #eeeeee solid;
	margin: 20px 30px 10px 10px;



}

#pub input{
	background-color:#222222;
	color: white;
	border: 1px #eeeeee solid;
	margin: 20px 30px 10px 10px;

}

#sub input{
	background-color:#222222;
	color: white;
```

```
        border: 1px #eeeeee solid;
        margin: 20px 30px 10px 10px;

}



#on{
        color: #ffffff;
        background-color: limegreen;
        width: 100px;
        height: 40px;
        padding: 0px;
}
#off{
        background-color: orangered;
        color: #ffffff;
        width: 100px;
        height: 40px;
        padding: 0px;
}

input{
        width: 300px;
        height: 30px;
        border-radius: 5px;
        padding-left: 10px;
}

button{
        margin: 20px 30px 10px 10px;
        border-radius: 5px;

}

select{
        background-color: #222222;
        color: #ffffff;
        width: 100px;
        height: 30px;
        padding-left: 10px;
        margin-left: 10px;
        border-radius: 5px;

}

#pub_button{
        color: #ffffff;
        background-color: limegreen;
        width: 100px;
        height: 40px;
        padding: 0px;
}

#sub_on_button{

        color: #ffffff;
        background-color: limegreen;
        width: 100px;
        height: 40px;
        padding: 0px;
    }

    #sub_off_button{
        color: #ffffff;
        background-color: orangered;
        width: 100px;
        height: 40px;
        padding: 0px;
    }
    #sub_qos{

        margin: 20px 30px 10px 10px;
    }

    th{
        height: 40px;
    }

    td{
        height: 40px;
        padding-left: 10px;
    }



</style>
<body>
```

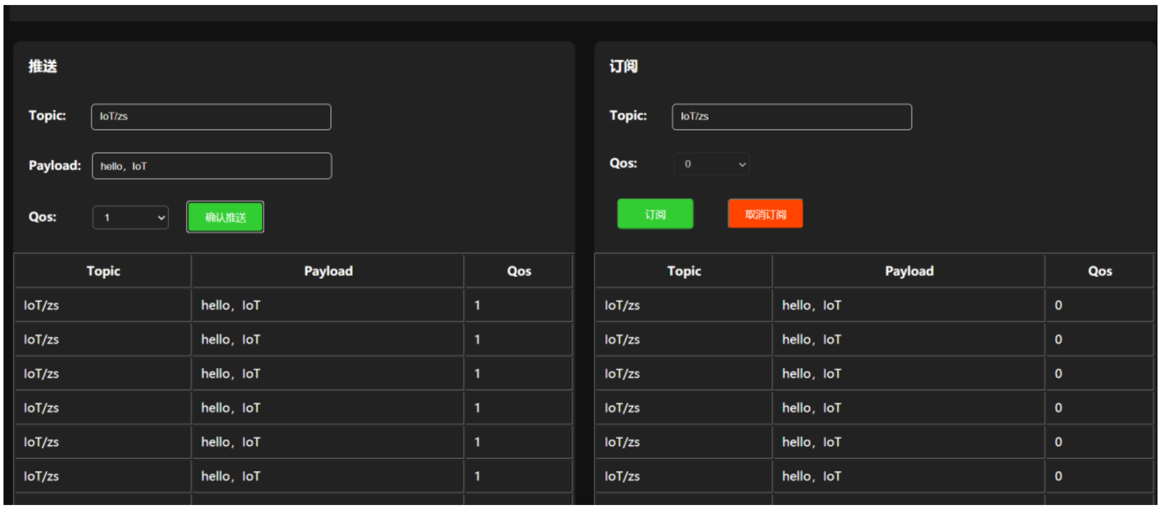
```
  <h2   style="color: white;padding-left: 20px;">WebSocket</h2>
  <div id="connection">
              <form id="connection_form" name="connection_form" method="post">
              <h3>连接</h3>
              <b>主机地址: </b><input type="text" value="47.100.136.15" id="host">
              <b>端口: </b><input type="text" value="8083" id="port">
              <b>Path: </b><input type="text" value="/mqtt" id="path">
              <br>
              <b>用 户 名: </b><input type="text" id="username">
              <b>密码: </b><input type="text" id="password">
              <br>
              <button type="button" onclick="connection() " id="on">连接</button>
              <button       type="button"              disabled="disabled"       id="off"
onclick="disconnect()">断开连接</button>

              <b >当前连接状态: </b>
              <b id="info" style="color: blue;">未连接</b>
              </form>
  </div>
  <br>
  <div  id="pub"  style="background-color:#222222;height:100%;width:49%;float:left;color:
white;border-radius: 10px;margin: 5px; ">
        <h3 style="padding-left:20px">推送</h3>
              <form style="padding-left:20px">
              <b>Topic:</b>&nbsp;&nbsp;&nbsp;&nbsp;
              <input type="text" id="pub_topic">
              <br>
              <b>Payload:</b>
              <input type="text" id="pub_payload">
              <br>
```

```
            <b>Qos:</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            <select name="qos" id="pub_qos" style="margin-bottom: 31px;">
                <option value="0" selected="">0</option>
                <option value="1">1</option>
                <option value="2">2</option>
            </select>&nbsp;&nbsp;
            <button type="button" id="pub_button" disabled="disabled"
onclick="pub()">确认推送</button>
        </form>
        <table width="100%" border="1" id="pub_Tbl">
            <tbody>
            <tr>
                    <th scope="col" name="topic">Topic</th>
                    <th scope="col" name="payload">Payload</th>
                    <th scope="col" name="qos">Qos</th>
            </tr>
                </tbody>
        </table>
  </div>
  <div id="sub" style="background-color:#222222;height:100%;width:49%;float:left;color:
white;border-right: 1px;border-radius: 10px;margin: 5px 0px 5px 20px;">
        <h3 style="padding-left:20px">订阅</h3>
        <form style="padding-left:20px">
            <b>Topic:</b>&nbsp;&nbsp;&nbsp;&nbsp;
            <input type="text" id="sub_topic">
            <br>
            <b>Qos:</b>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            <select name="pub_qos" id="sub_qos" >
                <option value="0" selected="">0</option>
                <option value="1">1</option>
                <option value="2">2</option>
            </select>&nbsp;&nbsp;
            <br>
            <button type="button" id="sub_on_button" disabled="disabled"
onclick="sub_on()">订阅</button>
            <button type="button" id="sub_off_button" disabled="disabled"
onclick="sub_off()">取消订阅</button>
        </form>
        <br>
        <table width="100%" border="1" id="sub_Tbl">
            <tbody>
            <tr>
                    <th scope="col">Topic</th>
                    <th scope="col">Payload</th>
                    <th scope="col">Qos</th>
            </tr>
        </tbody>
        </table>
  </div>



<script>
  //连接
  var options;
  var client;
  var connectioning=false;
  function connection() {
        if(host.value ==""||port.value==""||host.value ==" "||port.value==" "){
            window.alert("主机地址和端口不能为空");
            return;
        }
        options = {

            clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),
            username: username.value,
            password: password.value
        }
        console.log("ws://"+host.value+":"+port.value+path.value);

        try{
            client = mqtt.connect("ws://"+host.value+":"+port.value+path.value,options);
            document.getElementById ("on").setAttribute("disabled", "disabled");
            document.getElementById ("info").innerHTML =("连接中");
            document.getElementById ("info").style.color = 'orange';
            connectioning=true;
            setTimeout(function () {
                if(connectioning == true){
                    window.alert("连接超时");
                    document.getElementById ("info").innerHTML =("连接超时");
                    document.getElementById ("info").style.color = 'red';
                    document.getElementById ("on").removeAttribute("disabled");
                    connectioning = false;
                    client.end();
                }
            },5000);
            client.on('connect', function () {
                connectioning=false;

                document.getElementById ("info").innerHTML =("连接成功");
```

```
                    document.getElementById ("info").style.color = 'green';

                    document.getElementById ("off").removeAttribute("disabled");
                    document.getElementById ("pub_button").removeAttribute("disabled");
                    document.getElementById
("sub_on_button").removeAttribute("disabled");

                    document.getElementById ("host").setAttribute("disabled", "disabled");
                    document.getElementById ("port").setAttribute("disabled", "disabled");
                    document.getElementById ("path").setAttribute("disabled", "disabled");
                    document.getElementById          ("username").setAttribute("disabled",
"disabled");
                    document.getElementById          ("password").setAttribute("disabled",
"disabled");
                })
            }
            catch (err) {
                window.alert("连接失败!!!\n 错误信息："+err)

                document.getElementById ("info").innerHTML =("连接失败")
                document.getElementById ("info").style.color = 'red';
            }
        }
        function disconnect(){
            connectioning=false

            document.getElementById ("info").innerHTML =("未连接");
            document.getElementById ("info").style.color = 'blue';

            document.getElementById ("on").removeAttribute("disabled");
            document.getElementById ("off").setAttribute("disabled", "disabled");
            document.getElementById ("pub_button").setAttribute("disabled", "disabled");
            document.getElementById ("sub_on_button").setAttribute("disabled", "disabled");
            document.getElementById ("sub_off_button").setAttribute("disabled", "disabled");

            document.getElementById ("host").removeAttribute("disabled");
            document.getElementById ("port").removeAttribute("disabled");
            document.getElementById ("path").removeAttribute("disabled");
            document.getElementById ("username").removeAttribute("disabled");
            document.getElementById ("password").removeAttribute("disabled");

            client.end(true);
        }
        function pub() {
            if(pub_topic.value == "" ||pub_payload.value == ""||pub_topic.value == " "
||pub_payload.value == " "){
                window.alert("Topic 和  Payload 不能为空");
                return;
            }
            var pub_options={
                qos: parseInt(pub_qos.value)
            };
            try{
                client.publish(pub_topic.value,pub_payload.value,pub_options,function (e){
                    var oTr = document.getElementById("pub_Tbl").insertRow(1);
                    var aText = new Array();
                    aText[0] = document.createTextNode(pub_topic.value);
                    aText[1] = document.createTextNode(pub_payload.value);
                    aText[2] = document.createTextNode(pub_qos.value);
                    for (var index = 0; index < aText.length; index++) {
```

```
                var oTd = oTr.insertCell(index);
                oTd.appendChild(aText[index]);
            }
        });
    }
    catch (err) {
        window.alert("发送失败\n 错误信息:"+err)
    }
}
function sub_on() {
    console.log(client.messageIdToTopic)
    options = {

        clientId: 'mqttjs_' + Math.random().toString(16).substr(2, 8),
        username: username.value,
        password: password.value
    }
    console.log("ws://"+host.value+":"+port.value+path.value);
    if(sub_topic.value == ""||sub_topic.value == " "){
        window.alert("Topic 不能为空");
        return;
    }
    var sub_option = {
        qos: parseInt(sub_qos.value)
    }
    client.subscribe(sub_topic.value,sub_option,function (err,granted) {
        if (!err) {
            console.log(granted);
            document.getElementById ("sub_qos").setAttribute("disabled", "disabled");
            document.getElementById          ("sub_topic").setAttribute("disabled",
"disabled");
                    document.getElementById
("sub_off_button").removeAttribute("disabled");
                    document.getElementById      ("sub_on_button").setAttribute("disabled",
"disabled");

                } else {
                    window.alert("订阅失败\n 错误信息:" + err)
                }
            })
            client.on('message', function (topic, payload,packet){
                console.log(packet);
                var oTr = document.getElementById("sub_Tbl").insertRow(1);
                var aText = new Array();
                aText[0] = document.createTextNode(topic);
                aText[1] = document.createTextNode(payload);
                aText[2] = document.createTextNode(packet.qos);
                for (var index = 0; index < aText.length; index++) {
                    var oTd = oTr.insertCell(index);
                    oTd.appendChild(aText[index]);
                }
            })
    }
    function sub_off() {
            client.unsubscribe(sub_topic.value,function (err){
                if(err){
                    window.alert("取消订阅失败\n 错误原因:"+err);
                }
                document.getElementById ("sub_qos").removeAttribute("disabled");
                document.getElementById ("sub_topic").removeAttribute("disabled");
                document.getElementById ("sub_on_button").removeAttribute("disabled");
                document.getElementById            ("sub_off_button").setAttribute("disabled",
"disabled");

        })
    }
</script>
</body></html>
```

截图：



推送

Topic: IoT/zs

Payload: hello，IoT

Qos: 1

Topic	Payload	Qos
IoT/zs	hello，IoT	1
IoT/zs	hello，IoT	1
IoT/zs	hello，IoT	1
IoT/zs	hello，IoT	1
IoT/zs	hello，IoT	1
IoT/zs	hello，IoT	1

订阅

Topic: IoT/zs

Qos: 0

Topic	Payload	Qos
IoT/zs	hello，IoT	0
IoT/zs	hello，IoT	0
IoT/zs	hello，IoT	0
IoT/zs	hello，IoT	0
IoT/zs	hello，IoT	0
IoT/zs	hello，IoT	0

3. 订阅并解析来自自己完成的 WebSocket 客户端的 JSON 格式的消息，并将内容写入数据表中；

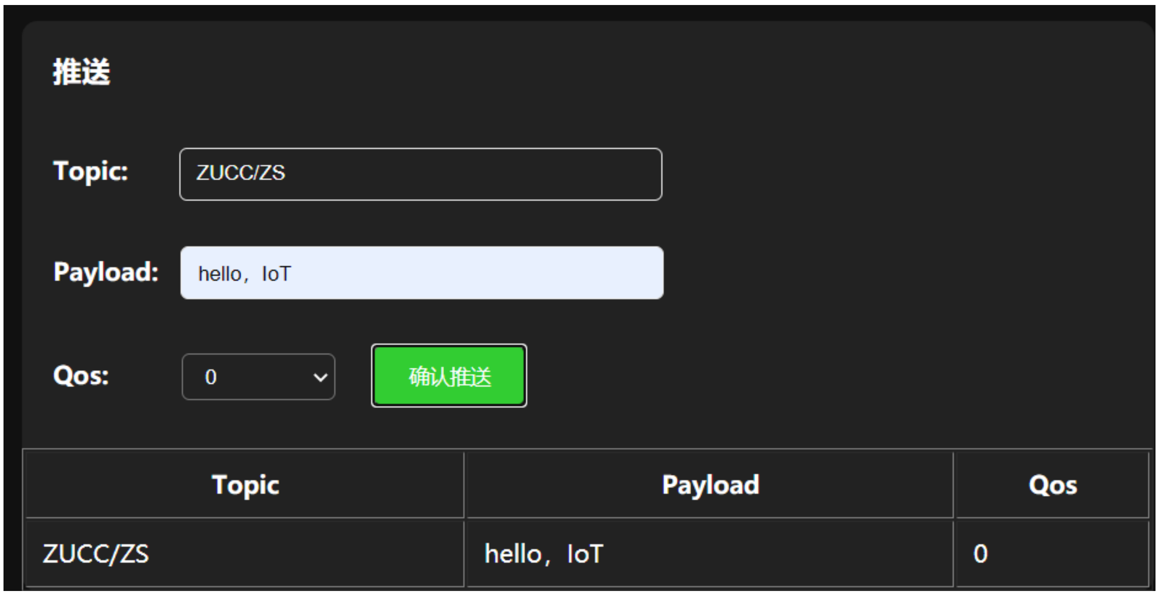
（WebSocket Topic 主题格式：**ZUCC/CJP**，将 **CJP** 改成自己姓名的缩写）

代码：

```java
package zs;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import com.alibaba.fastjson.JSONObject;

public class SaveMysql {
    private static String driver="com.mysql.jdbc.Driver";
    private static String
```

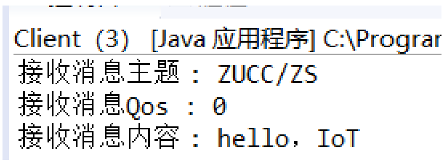
```
url="jdbc:mysql://47.100.136.15:3306/mqtt?serverTimezone=UTC";
    private static String user="root";
    private static String password="Sss991126/";
    Connection conn=null;
    Statement stmt=null;
    ResultSet rs=null;


    public void savedate2(String payload) {
        JSONObject equipment = new JSONObject();

        java.sql.Timestamp time= new 
java.sql.Timestamp(System.currentTimeMillis());
        try{

            equipment = JSONObject.parseObject(payload);
        }
        catch (Exception e){
            e.printStackTrace();
            System.out.println("Json格式错误");
            return ;
        }
        String sql = "insert into 
IoT_test(Sid,Sname,Ename,Humidity,Temperature,Timesite,Username)"+"values('
"+equipment.getString("sid")+"','"

+equipment.getString("sname")+"','"+equipment.getString("Ename")+"','"+equi
pment.getDouble("humidity")+"','"+equipment.getDouble("temperature")
                +"','"+time+"','"+equipment.getString("clientid")+"');";
        try {
            Class.forName(driver);
            conn=DriverManager.getConnection(url,user,password);
            stmt=conn.createStatement();
            stmt.executeUpdate(sql);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            try {
                if(stmt!=null) stmt.close();
                if(conn!=null) conn.close();
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }

    }

    public void savedate(String topic,int Qos,String payload) {

        java.sql.Timestamp timestamp=new
java.sql.Timestamp(System.currentTimeMillis());
        String sql = "insert into
mqtt(Topic,Qos,payload,Timestamp)"+"values('"+topic+"','"+Qos+"','"+payload
+"','"+timestamp+"');";
        try {
            Class.forName(driver);
            conn=DriverManager.getConnection(url,user,password);
            stmt=conn.createStatement();
            stmt.executeUpdate(sql);
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        finally {
            try {
                if(stmt!=null) stmt.close();
                if(conn!=null) conn.close();
            }
            catch(Exception e) {
                e.printStackTrace();
            }
        }
    }


}
```

截图:

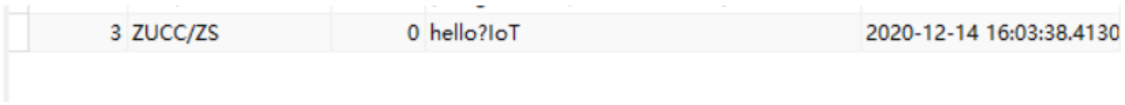


Topic: ZUCC/ZS
Payload: hello，IoT
Qos: 0

Topic	Payload	Qos
ZUCC/ZS	hello，IoT	0



```
Client (3) [Java 应用程序] C:\Prograr
接收消息主题 : ZUCC/ZS
接收消息Qos : 0
接收消息内容 : hello，IoT
```



3	ZUCC/ZS	0	hello?IoT	2020-12-14 16:03:38.4130

4. 采集 ESP8266 的温度、湿度、时间等数据，构建 JSON 格式的消息，并写入到 MySQL 数据库中。

参考字段：

序号	学号	姓名	终端名称	湿度	湿度	时间	用户名

代码：

```
#include <ESP8266WiFi.h>

#include "Adafruit_MQTT.h"

#include "Adafruit_MQTT_Client.h"

#define WLAN_SSID       "Mi 10000 Ultra"

#define WLAN_PASS       "88888888"

#define AIO_SERVER       "47.100.136.15"

#define AIO_SERVERPORT      1883

#define AIO_USERNAME      ""
```

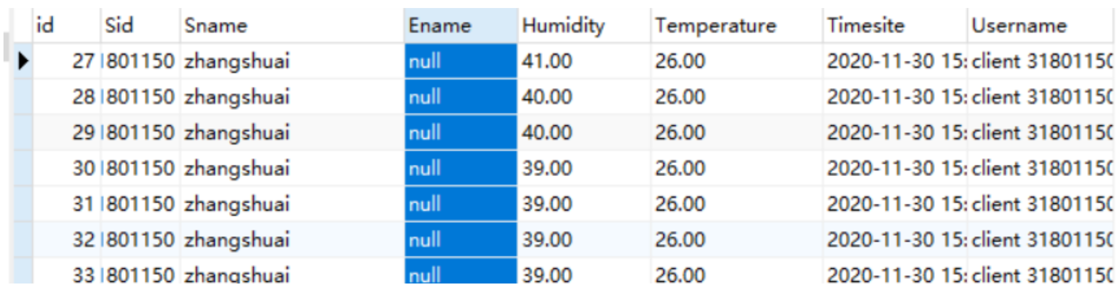
```
#define AIO_KEY              ""
#include"SSD1306Wire.h"
SSD1306Wire display(0x3c,2,14);
#include<dht11.h>
#include<ArduinoJson.h>

dht11 DHT11;
// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;
// or... use WiFiClientSecure for SSL
//WiFiClientSecure client;

StaticJsonDocument<200> doc;
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
// Setup a feed called 'photocell' for publishing.
// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish photocell = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "toZS/1150");

// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "toZS/1150");

// Bug workaround for Arduino 1.6.6, it seems to need a function declaration
// for some reason (only affects ESP8266, likely an arduino-builder bug).
void MQTT_connect();
void setup() {


Serial.begin(115200);
delay(10);
  display.init();


Serial.println(F("Adafruit MQTT demo"));


// Connect to WiFi access point.
Serial.println();
Serial.println();
Serial.print("Connecting to ");
Serial.println(WLAN_SSID);


WiFi.begin(WLAN_SSID, WLAN_PASS);
while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.println();


Serial.println("WiFi connected");
Serial.println("IP address: "); Serial.println(WiFi.localIP());


// Setup MQTT subscription for onoff feed.
mqtt.subscribe(&onoffbutton);
}
uint32_t x=0; void loop() {
        int chk = DHT11.read(5);
// Ensure the connection to the MQTT server is alive (this will make the first
//    connection   and  automatically  reconnect  when   disconnected).   See   the
```

```
MQTT_connect
  // function definition further below.
  MQTT_connect();

  // this is our 'wait for incoming subscription packets' busy subloop
  // try to spend your time here

  Adafruit_MQTT_Subscribe *subscription;
  while ((subscription = mqtt.readSubscription(5000))) { if (subscription == &onoffbutton)
{
    Serial.print(F("Got: "));
    Serial.println((char *)onoffbutton.lastread);

    display.flipScreenVertically();
    display.clear();
    display.drawString(0,10,"topic: toZS/1150");
    display.drawString(0,20,(char *)onoffbutton.lastread);
   // display.display();
    delay(2000);
  }
  }
  // Now we can publish stuff!
  Serial.print(F("\nSending photocell val ")); Serial.print(x);
  Serial.print("...");

  String output;

  doc["sid"] = "31801150";
  doc["sname"] = "zhangshuai";
  doc["ename"] = "test1";
```

```
doc["clientid"] = "client 31801150";
doc["humidity"] = (float)DHT11.humidity;
doc["temperature"] = (float)DHT11.temperature;
serializeJson(doc ,output);




if (! photocell.publish(output.c_str())) { Serial.println(F("Failed"));
} else {
Serial.println(F("OK!"));

   display.flipScreenVertically();
   display.clear();
   display.drawString(0,10,"topic: iot/1");
   display.drawString(0,20,String(x));
//    display.display();
   delay(2000);
}

// ping the server to keep the mqtt connection alive
// NOT required if you are publishing once every KEEPALIVE seconds
/*
if(! mqtt.ping()) { mqtt.disconnect();
}
*/
}

// Function to connect and reconnect as necessary to the MQTT server.
// Should be called in the loop function and it will take care if connecting.
void MQTT_connect() {
```

```
int8_t ret;


// Stop if already connected.
if (mqtt.connected()) {
return;
}
Serial.print("Connecting to MQTT... "); uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
   Serial.println(mqtt.connectErrorString(ret));
Serial.println("Retrying MQTT connection in 5 seconds..."); mqtt.disconnect();
delay(5000);   // wait 5 seconds retries--;
if (retries == 0) {
// basically die and wait for WDT to reset me
while (1);
}
}
Serial.println("MQTT Connected!");
}
```

截图:



id	Sid	Sname	Ename	Humidity	Temperature	Timesite	Username
27	1801150	zhangshuai	null	41.00	26.00	2020-11-30 15:	client 3180115
28	1801150	zhangshuai	null	40.00	26.00	2020-11-30 15:	client 3180115
29	1801150	zhangshuai	null	40.00	26.00	2020-11-30 15:	client 3180115
30	1801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115
31	1801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115
32	1801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115
33	1801150	zhangshuai	null	39.00	26.00	2020-11-30 15:	client 3180115