

首页

留言本

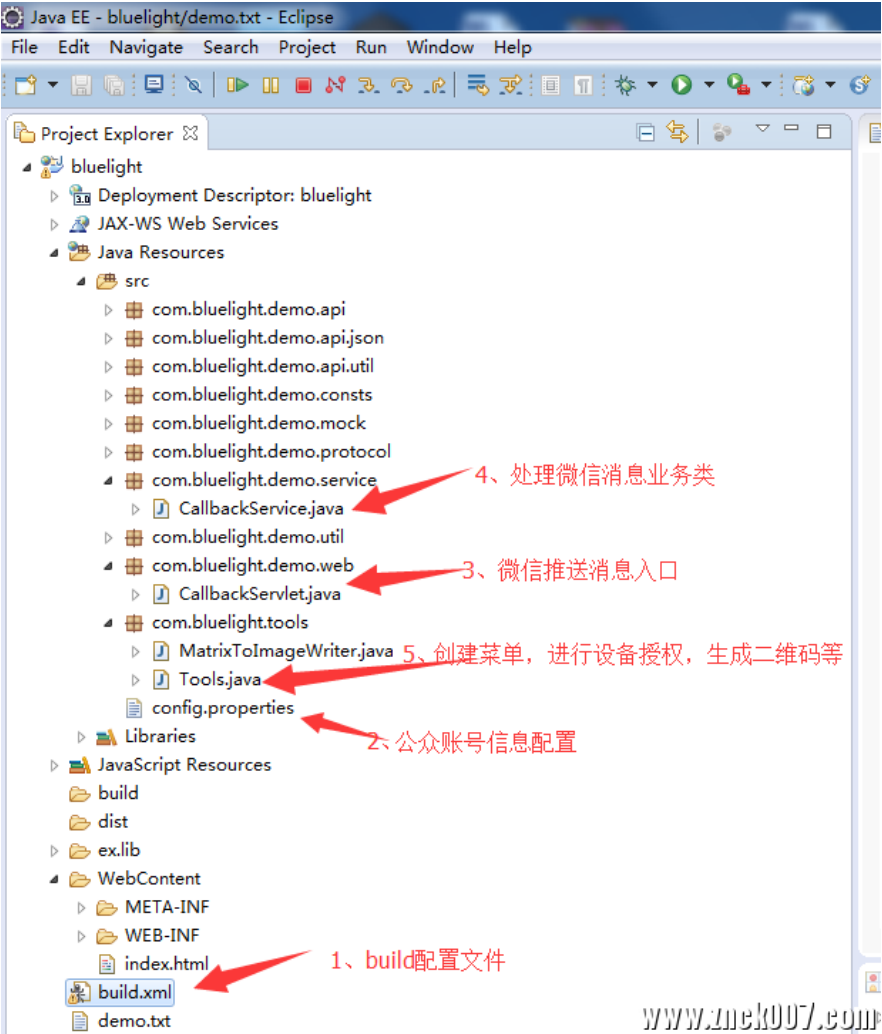
07

201905

微信硬件开发系列教程04-微信公众号开发DEMO(airkiss/airsync)

这篇讲解对微信公众号开发DEMO分析，主要的目的是了解它的原理和微信API的使用。然后进行修改，实现我们想要的功能，如：菜单的创建、信息交互、跳转到自定义HTML5页面等等。

一、目录结构讲解。



- 展开目录后，按照1、2、3、4、5的顺序讲解：
- 1、web.xml网站配置（上面图片标错了，是在WEB-INF未展来的目录里的web.xml）
 - 2、公众账号信息配置。
 - 3、微信推送消息的入口。
 - 4、处理微信消息的业务类。
 - 5、相关微信API的使用方法，创建菜单，进行设备授权，生成二维码等。

二、web.xml（程序引导配置）

网站分类

- 未分类
- 智能家居DIY
- 微信硬件开发
- WIFI作品DIY
- APP开发
- Zigbee开源智能家居
- 百元智能家居
- 新手求助区

最新留言

源代码应该都还是能下载的，现在...
硬件卖吗？能否提供有偿技术指导...
了不起，有不同场景时，能请你指...
博主加油你说拖延症晚期症状是什...
非常感谢by <https://92huayi.com/>...
博主加油by <https://www.daantu.c...>...
感谢您的付出感谢您。by <https://d...>...
加油加油by <https://www.92huayi...>...
书已经到手，但在此处无法下载视...
哪里可以买硬件

控制面板

您好，欢迎访问网站！
[登录后台](#) [查看权限](#)

文章归档

- 2019年5月 (43)
- 2019年4月 (23)

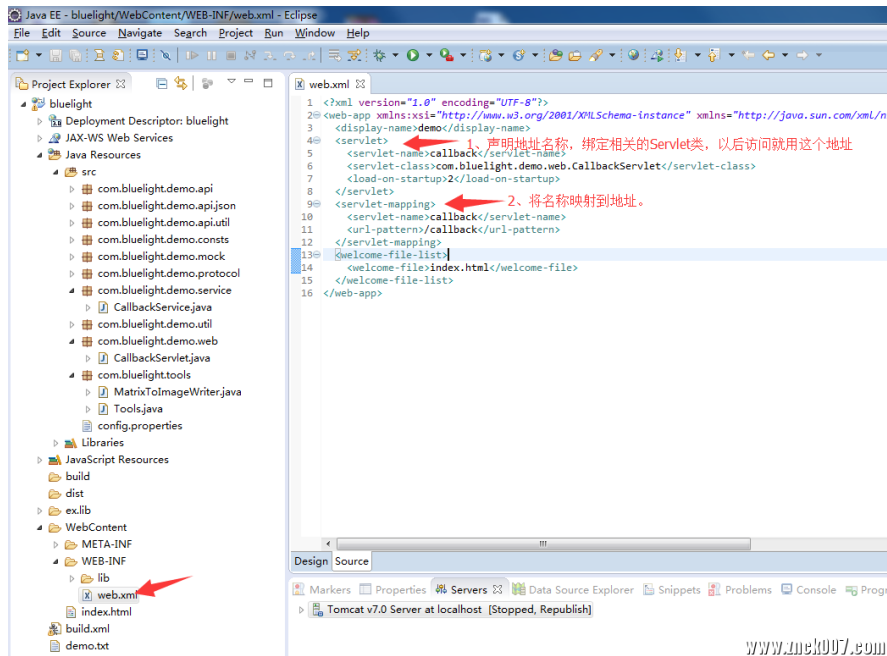
网站收藏

- Z-Blog应用中心
- Z-Blog官方微博
- ZBlogger社区

友情链接

- Z-Blog on Github
- Z-Blog主机



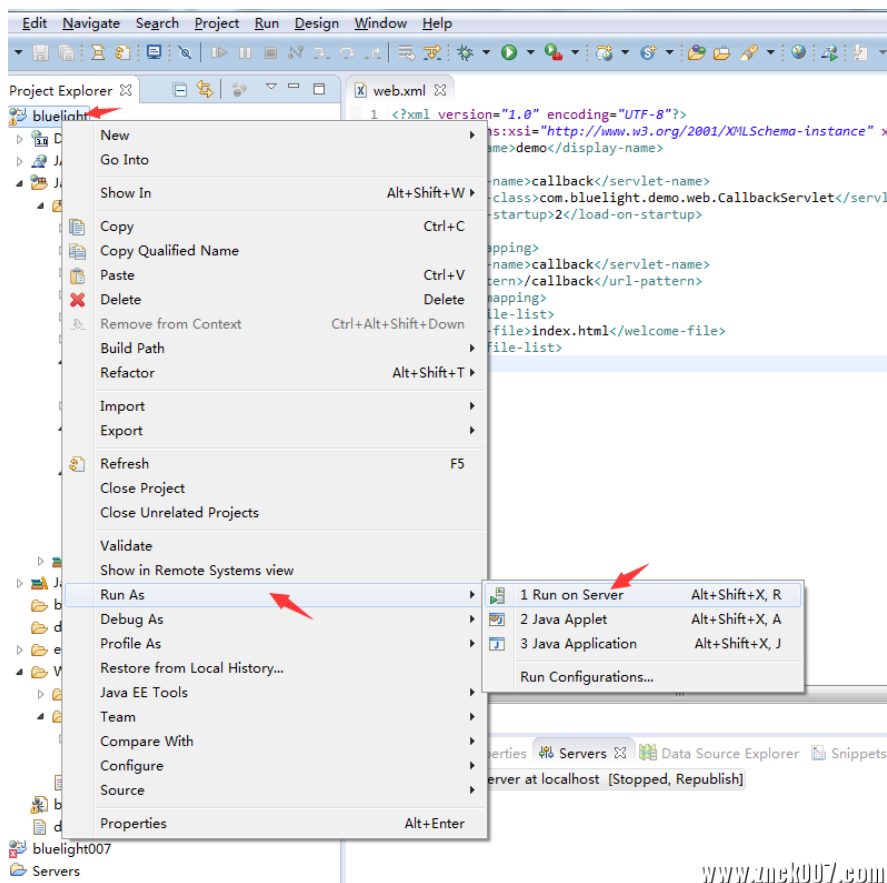


www.znck007.com

双击web.xml，就会看到网站配置的内容

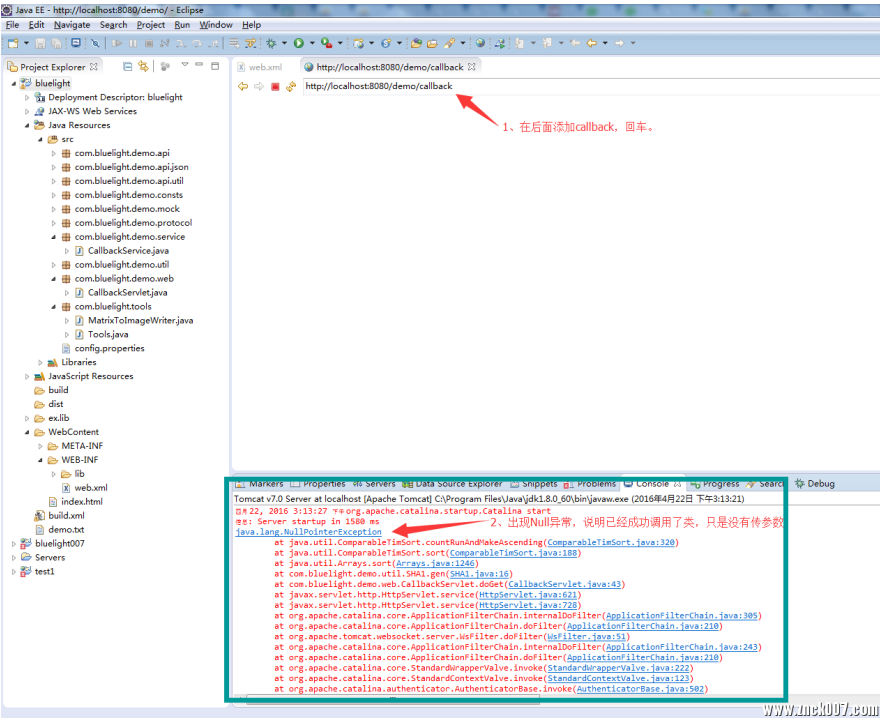
- 1、声明地址名称，绑定相关的Servlet类，以后访问就用这个地址。
- 2、将名称映射到地址。

这样就能成功访问：http://你的域名/callback来访问com.bluelight.demo.web.CallbackServlet类了。



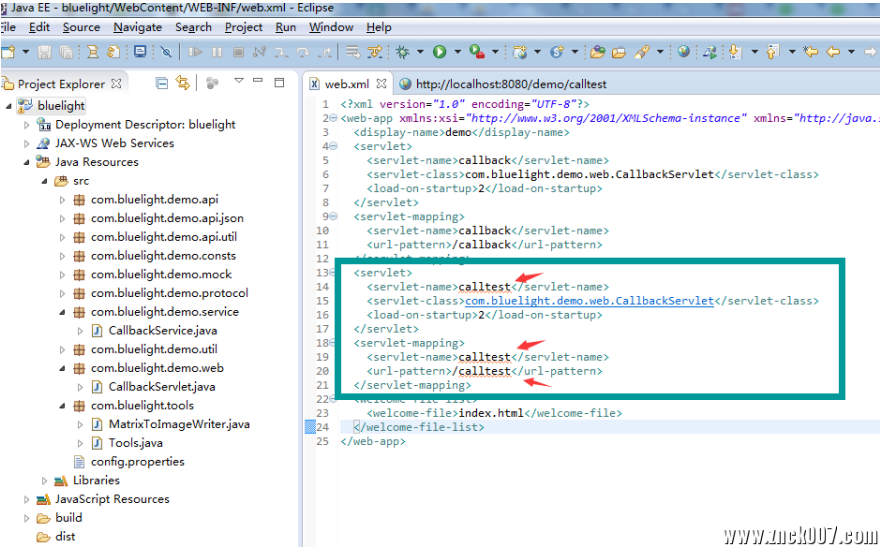
www.znck007.com

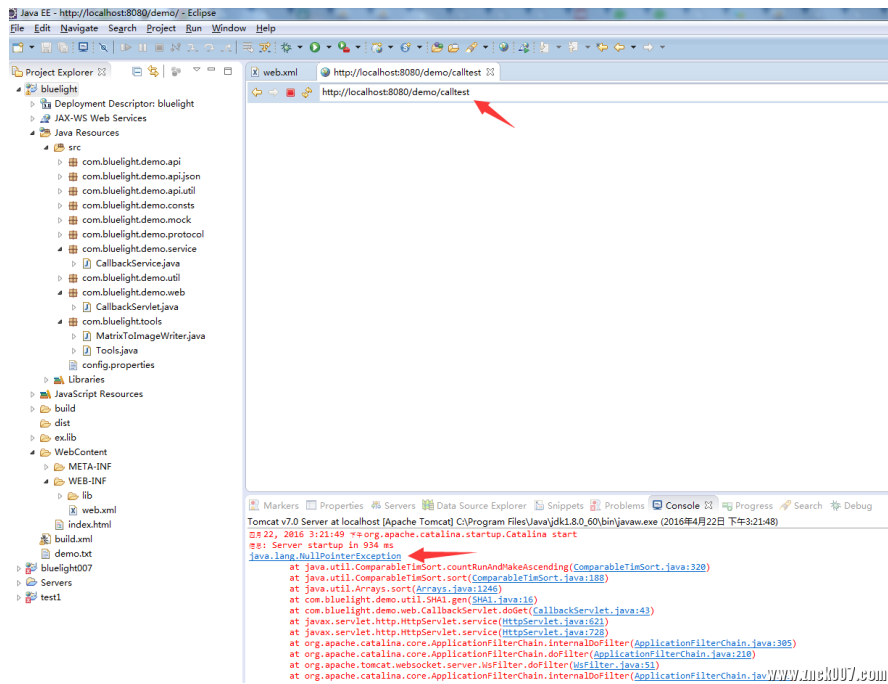
我们运行，右击bluelight->Run As->Run Server，然后Finish。



- 1、在后面添加callback，回车。
- 2、出现Null异常，说明已经成功调用了类，只是没有传参数。

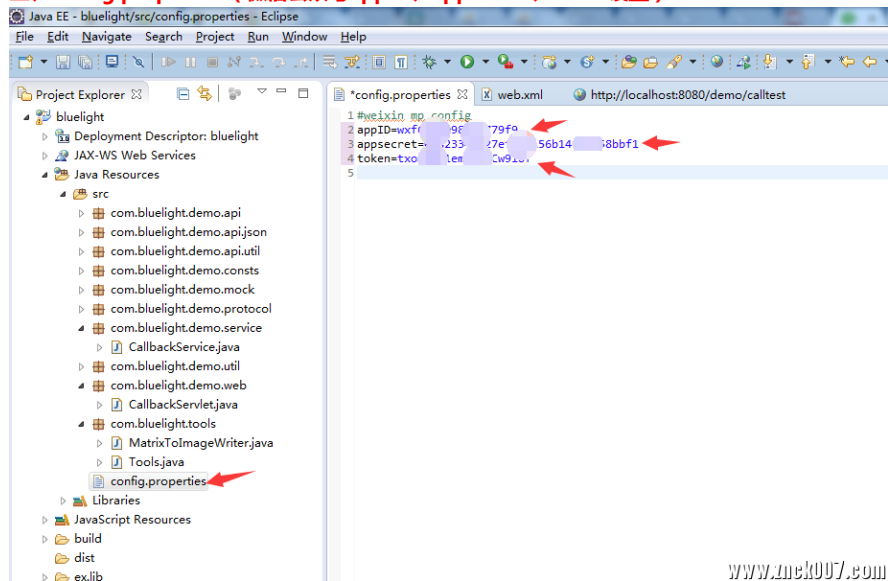
那我们如何添加一个访问类地址呢？





重新运行，然后访问<http://localhost:8080/demo/calltest>，成功调用了类文件。
这里还有一种是jsp，可以直接放到WebContent目录里，直接访问文件名就可以，不需要做映射。

三、config.properties (微信公众号appID、appsecret、taken设置)



双击config.properties，打开后appID、appsecret、taken都是空的，我们要复制进去。

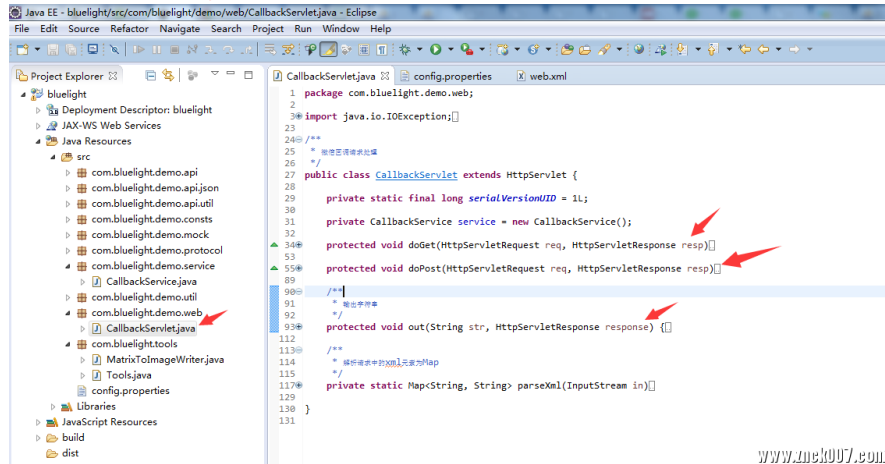


登陆公众号开发测试网址：

<http://mp.weixin.qq.com/debug/cgi-bin/sandboxinfo?action=showinfo&t=sandbox/index>

直接复制appID、appsecret填写，URL是我们之后要教大家申请的服务器地址，taken自定义一个就可以了。

四、CallbackServlet.java（入口类讲解）



打开类，主要看到三个类：doGet、doPost、out函数。

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    try {
        // 开发者接入验证
        String timestamp = req.getParameter("timestamp");
        String nonce = req.getParameter("nonce");
        String signature = req.getParameter("signature");
        String echostr = req.getParameter("echostr");

        if (signature.equals(SHA1.gen(WxConfig.TOKEN, timestamp, nonce))) {
            out(echostr, resp);
        } else {
            out("", resp);
        }
    } catch (Throwable e) {
        e.printStackTrace();
        out("", resp);
    }
}
```

在这里的doGet函数，基本不做事，只接收到传来的值，然后直接就返回空的值回去了，如果验证不通过就返回错误而以。

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
    try {
        // 编码格式
        req.setCharacterEncoding("UTF-8");

        // 验证签名
        String timestamp = req.getParameter("timestamp");
        String nonce = req.getParameter("nonce");
        String signature = req.getParameter("signature");
        if (!signature.equals(SHA1.gen(WxConfig.TOKEN, timestamp, nonce))) {
            out("", resp);
            return;
        }

        // 解析XML
        Map<String, String> reqMap = parseXml(req.getInputStream());
        System.out.println("reqMap=" + reqMap);

        // 处理请求
        String xmlStr = service.handle(reqMap);

        System.out.println("xmlStr=" + xmlStr);

        // null 转为空字符串
        xmlStr = xmlStr == null ? "" : xmlStr;

        out(xmlStr, resp);
    } catch (Throwable e) {
        e.printStackTrace();
        // 异常时响应空串
        out("", resp);
    }
}
```

doPost函数，多了解xml，然后使用CallbackService业务类进行处理（这才是关键的地方），然后调用out函数才返回给微信。

```

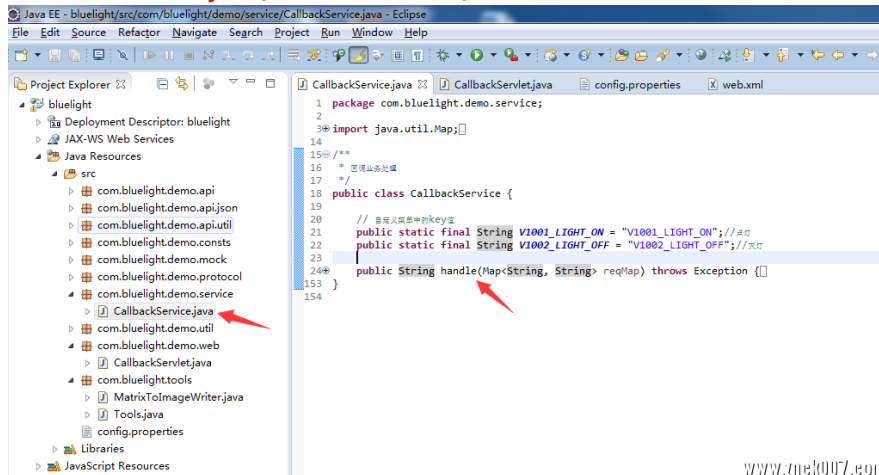
90  /**
91  * 输出字符串
92  */
93  protected void out(String str, HttpServletResponse response) {
94      Writer out = null;
95      try {
96          response.setContentType("text/xml;charset=UTF-8");
97          out = response.getWriter();
98          out.append(str);
99          out.flush();
100     } catch (IOException e) {
101         // ignore
102     } finally {
103         if (out != null) {
104             try {
105                 out.close();
106             } catch (IOException e) {
107                 // ignore
108             }
109         }
110     }
111 }

```

www.znck007.com

out函数，使用了Writer类，然后输出返回，这样微信就能收到啦。

五、CallbackService.java（微信消息业务类讲解）



www.znck007.com

在入口类的doPost函数里，调用了这个业务类，那我们来——分析它。
这个类就只有一个事件处理函数handle

```

0 package com.bluelight.demo.service;
0
1 import java.util.Map;
0
0 import org.apache.commons.codec.binary.Base64;
2
0 import com.bluelight.demo.api.DeviceApi;
0 import com.bluelight.demo.api.MpApi;
3 import com.bluelight.demo.consts.MsgType;
0 import com.bluelight.demo.consts.XmlResp;
0 import com.bluelight.demo.mock.DBMock;
4 import com.bluelight.demo.protocol.BlueLight;
0 import com.bluelight.demo.protocol.BlueLight.CmdId;
0
5 /**

```

```

0  * 回调业务处理
0  */
6 public class CallbackService {
0
0      // 自定义菜单中的key值
7      public static final String V1001_LIGHT_ON = "V1001_LIGHT_ON";//点灯
0      public static final String V1002_LIGHT_OFF = "V1002_LIGHT_OFF";//灭灯
0
8      public String handle(Map<String, String> reqMap) throws Exception {
0          String msgType = reqMap.get("MsgType");
0          String fromUser = reqMap.get("FromUserName");
9          String toUser = reqMap.get("ToUserName");
0
1          // 针对不同类型的消息和事件进行处理
0
0          // 文本消息
1          if (MsgType.TEXT.equals(msgType)) {
1              // 可以在此处进行关键字自动回复
0              String content = "收到文本消息: " + reqMap.get("Content");
1              return XmlResp.buildText(fromUser, toUser, content);
2          }
0
1          // 基础事件推送
3          if (MsgType.EVENT.equals(msgType)) {
0              String event = reqMap.get("Event");
1              // 关注公众号
4              if (MsgType.Event.SUBSCRIBE.equals(event)) {
0                  // 回复欢迎语
1                  return XmlResp.buildText(fromUser, toUser, "欢迎
0 关注蓝牙灯泡demo测试公众号!");
1              }
6              // 菜单点击事件
0              if (MsgType.Event.CLICK.equals(event)) {
1                  // 根据key值判断点击的哪个菜单
7                  String eventKey = reqMap.get("EventKey");
0                  // 点灯/灭灯
1                  if (V1001_LIGHT_ON.equals(eventKey)
8                      || V1002_LIGHT_
0FF.equals(eventKey)) {
1                      //是否点灯操作
9                      boolean open = V1001_LIGHT_ON.eq
0uals(eventKey);
2
0                      // 根据 fromUserName 获取绑定的
0 信息
2                      Map<String, String> boundInfo =
1DBMock.queryBoundInfo(fromUser);
0
2                      // 未绑定
2                      if (boundInfo == null) {
0                          return XmlResp.b
2uildText(fromUser, toUser, "未绑定");
0                      }
2
4                      String deviceType = boundInfo.g
0et("deviceType");
2                      String deviceID = boundInfo.get
5("deviceID");
0                      String openID = boundInfo.get(
2"openID");
6
0                      // 构造设备消息

```

```

0                                     CmdId cmdId = open ? BlueLight.
2 CmdId.OPEN_LIGHT_PUSH : BlueLight.CmdId.CLOSE_LIGHT_PUSH;
7                                     byte[] respRaw = BlueLight.buil
0 d(cmdId, null, (short)0).toBytes();
2                                     // Base64编码
8                                     final String content = Base64.en
0 codeBase64String(respRaw);
2
9                                     // 推送消息给设备
0                                     DeviceApi.transMsg(deviceType,
3 deviceID, openID, content);
0
0                                     // 回复
3                                     boolean debug = true;
1                                     if(debug){
0                                     // 返回调试信息
3                                     String debugTex
2 t = "已发送" + (open ? "点灯" : "灭灯") + "消息: " + "deviceID为" + deviceID + ",设备消息为" + content;
0                                     return XmlResp.b
3 uildText(fromUser, toUser, debugText);
0                                     }else{
3                                     return "";
4                                     }
0                                     }
3                                     }
5                                     }
0
3                                     // 设备消息或事件
6                                     if (MsgType.DEVICE_EVENT.equals(msgType)
0                                     || MsgType.DEVICE_TEXT.equals(msgType)) {
3                                     String reqContent = reqMap.get("Content");
7                                     String deviceType = reqMap.get("DeviceType");
0                                     String deviceID = reqMap.get("DeviceID");
3                                     String sessionID = reqMap.get("SessionID");
8                                     final String openID = reqMap.get("OpenID");
0                                     // 设备事件推送
3                                     if (MsgType.DEVICE_EVENT.equals(msgType)) {
9                                     String event = reqMap.get("Event");
0                                     // 绑定/解绑事件
4                                     if (MsgType.DeviceEvent.BIND.equals(event)
0                                     || MsgType.Devi
0 ceEvent.UNBIND.equals(event)) {
1                                     // 存储用户和设备的绑定关系
0                                     if(MsgType.DeviceEvent.BIND.equ
4 als(event)){
2                                     DBMock.saveBoun
0 dInfo(reqMap);
4                                     }else{
3                                     DBMock.removeBo
0 undInfo(reqMap.get("FromUserName"));
4                                     }
4                                     // 设备绑定/解绑事件可以回复空
0 包体
4                                     return "";
5                                     }
0                                     }
4                                     // 收到设备消息
6                                     if (MsgType.DEVICE_TEXT.equals(msgType)) {
0                                     // Base64解码
4                                     byte[] reqRaw = Base64.decodeBase64(reqConten
7 t);
0                                     // 反序列化

```



```

0         BlueLight lightReq = BlueLight.parse(reqRaw);
4
8         // 逻辑处理
0         // demo中 推送消息给用户微信
4         String reqText = lightReq.body;
9         System.out.println("recv text:" + reqText);
0         String transText = "收到设备发送的数据: ";
5
0
0         byte[] reqTextRaw = reqText.getBytes("UTF-8");
5         if (reqTextRaw.length > 0 && reqTextRaw[reqTextR
1         aw.length - 1] == 0) {
0
0             // 推送给微信用户的内容去掉末尾
5             的反斜杠零'\0'
2
2             transText = transText + new Stri
0         ng(reqTextRaw, 0, reqTextRaw.length - 1, "UTF-8");
5
5             } else{
3
3             transText = transText + reqTex
0         t;
5
5             }
4
0         // 推送文本消息给微信
5         MpApi.customSendText(openID, transText);
5
0
0         // demo中 回复 收到的内容给设备
5         BlueLight lightResp = BlueLight.build(BlueLigh
6         t.CmdId.SEND_TEXT_RESP, reqText, lightReq.head.seq);
0
0         // 序列化
5         byte[] respRaw = lightResp.toBytes();
7         // Base64编码
0         String respCon = Base64.encodeBase64String(resp
5         Raw);
8
0
0
5         // 设备消息接口必须回复符合协议的xml
9         return XmlResp.buildDeviceText(toUser, fromUser,
0         deviceType, deviceID, respCon, sessionID);
6
6             }
0
0
0
0         // 未处理的情况返回空字符串
6         return "";
1
0
0         }
6    }
2
0
6
3
0
6
4
0
6
5
0
6
6
0
6
7
0
6
8

```

0
6
9
0
7
0
0
7
1
0
7
2
0
7
3
0
7
4
0
7
5
0
7
6
0
7
7
7
0
7
8
0
7
9
0
8
0
0
8
1
0
8
2
0
8
3
0
8
4
0
8
5
0
8
6
0
8
7
0
8
8
0
8
9

0
9
0
0
9
1
0
9
2
0
9
3
0
9
4
0
9
5
0
9
6
0
9
7
0
9
8
0
9
9
1
0
0
1
0
1
1
1
0
2
1
0
3
1
0
4
1
0
5
1
0
6
1
0
7
1
0
8
1
0
9
1
1
0

1
1
1
1
1
1
2
1
1
3
1
1
4
1
1
5
1
1
6
1
1
7
1
1
8
1
1
9
1
2
2
0
1
2
1
1
2
2
2
1
2
3
1
2
4
1
2
5
1
2
6
1
2
7
1
2
8
1
2
9
1
3
0
1
3
1

1
3
2
1
3
3
1
3
4
1
3
5
1
3
6
1
3
7
1
3
8
1
3
9
1
4
0
1
4
1
1
4
2
1
4
3
1
4
4
1
4
5
1
4
6
1
4
7
1
4
8
1
4
9
1
5
0
1
5
1
1
5
2

1
5
3

代码主要针对不同类型的消息，进行处理，如TEXT，EVENT，DEVICE_EVENT，DEVICE_TEXT。

TEXT是纯文本发过来，含表情。

EVENT是关注公众号，取消公众号，菜单点击等等，

DEVICE_EVENT是硬件的消息，DEVICE_TEXT就是硬件发送过来的文本（一般用文本交互就可以了）

```
// 消息点击事件
if (MsgType.Event.CLICK.equals(event)) {
    // 根据key值判断点击的那个消息
    String eventKey = reqMap.get("EventKey");
    // 开关/关灯
    if (V1001_LIGHT_ON.equals(eventKey)
        || V1002_LIGHT_OFF.equals(eventKey)) {
        // 开关灯操作
        boolean open = V1001_LIGHT_ON.equals(eventKey); // 开和关

        // 根据 fromUserName 获取绑定的信息
        Map<String, String> boundInfo = DBMock.queryBoundInfo(fromUser); // 获取绑定的设备

        // 未绑定
        if (boundInfo == null) {
            return XmlResp.buildText(fromUser, toUser, "未绑定");
        }

        String deviceType = boundInfo.get("deviceType"); // 获取硬件信息，有这些就可以交互了。
        String deviceID = boundInfo.get("deviceID");
        String openID = boundInfo.get("openID");

        // 构造设备消息
        CmdId cmdId = open ? BlueLight.CmdId.OPEN_LIGHT_PUSH : BlueLight.CmdId.CLOSE_LIGHT_PUSH;
        byte[] respRaw = BlueLight.build(cmdId, null, (short)0).toBytes();
        // Base64编码
        final String content = Base64.encodeBase64String(respRaw);

        // 构造消息给设备
        DeviceApi.transMsg(deviceType, deviceID, openID, content); // 发信息给设备

        // 日志
        boolean debug = true;
        if(debug){
            // 构造调试信息
            String debugText = "设备" + (open ? "开灯" : "关灯") + "消息: " + "deviceID为" + deviceID + ",设备消息为" + content;
            return XmlResp.buildText(fromUser, toUser, debugText);
        }else{
            return "";
        }
    }
}
```

www.znck007.com

在菜单点击事件里，实现了开和关，这里获取硬件的信息，前提是先扫二维码绑定，这个之后讲硬件时会讲到的。

```
// 设备事件推送
if (MsgType.DEVICE_EVENT.equals(msgType)) {
    String event = reqMap.get("Event");
    // 绑定/解绑事件
    if (MsgType.DeviceEvent.BIND.equals(event)
        || MsgType.DeviceEvent.UNBIND.equals(event)) {
        // 存储用户和设备的绑定关系
        if(MsgType.DeviceEvent.BIND.equals(event)){
            DBMock.saveBoundInfo(reqMap); // 绑定
        }else{
            DBMock.removeBoundInfo(reqMap.get("FromUserName")); // 解绑
        }
        // 设备绑定/解绑事件可以返回空包体
        return "";
    }
}
```

www.znck007.com

在设备事件里，如果是绑定事件和解绑事件，就进行临时的保存和删除。

```

// 收到设备消息
if (MsgType.DEVICE_TEXT.equals(msgType)) {
    // Base64解码
    byte[] reqRaw = Base64.decodeBase64(reqContent);
    // 反序列化
    BlueLight lightReq = BlueLight.parse(reqRaw);
    // 逻辑处理
    // demo中推送消息给用户微信
    String reqText = lightReq.body;
    System.out.println("recv text:" + reqText);
    String transText = "收到设备发送的数据: ";

    byte[] reqTextRaw = reqText.getBytes("UTF-8");
    if (reqTextRaw.length > 0 && reqTextRaw[reqTextRaw.length - 1] == 0) {
        // 推送给微信用户的内容表结束的标识符是'\0'
        transText = transText + new String(reqTextRaw, 0, reqTextRaw.length - 1, "UTF-8");
    } else {
        transText = transText + reqText;
    }

    // 推送文本消息给微信
    MpApi.customSendText(openID, transText);

    // demo中回复收到的内容给设备
    BlueLight lightResp = BlueLight.build(BlueLight.CmdId.SEND_TEXT_RESP, reqText, lightReq.head.seq);
    // 序列化
    byte[] respRaw = lightResp.toBytes();
    // Base64编码
    String respCon = Base64.encodeBase64String(respRaw);

    // 设备消息接口必须返回符合协议的XML
    return XmlResp.buildDeviceText(toUser, fromUser, deviceType, deviceID, respCon, sessionID);
}

```

进行Base64解码

解析数据

打印解析收到的文本

返回信息到微信给用户看

生成指定格式信息

编码

发送信息给设备

www.znck007.com

这里处理收到设备的信息处理。

[Java] 纯文本查看 复制代码

```

?
// 文本消息
1
2
3
4p.get("Content");
5if(reqMap.get("Content").equals("你好")){
6    content = "你也好，你全家都好。";
7}
8
9content);
return XmlResp.buildText(fromUser, toUser,
content);
}

```

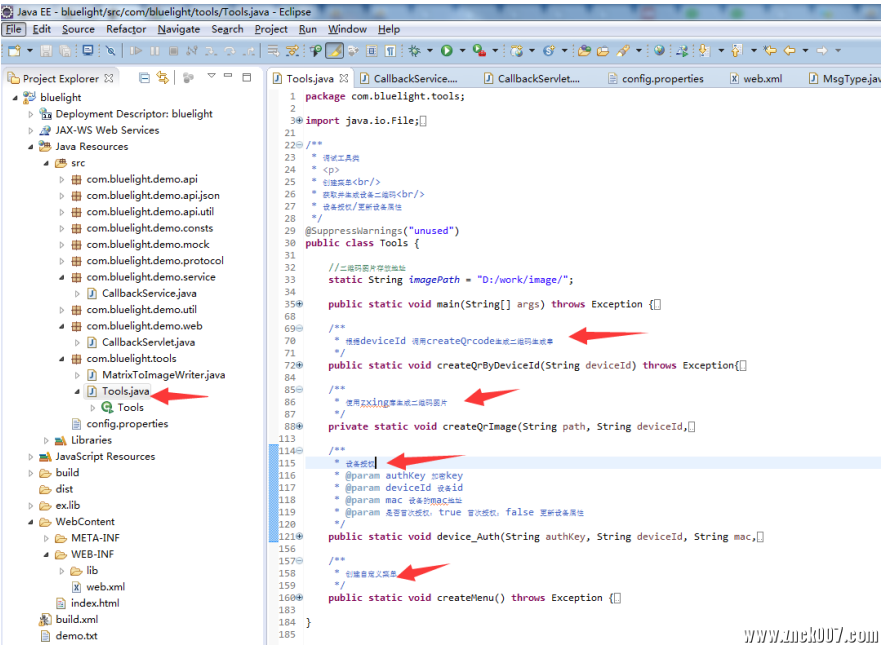
这里我们做一下修改，添加一段代码，如果用户发送“你好”两字，那么我直接回复他：你也好，你全家都好。

```

if(content.equals("你好")){
content = "你也好，你全家都好。";
}

```

六、Tools.java（操作微信API类讲解）

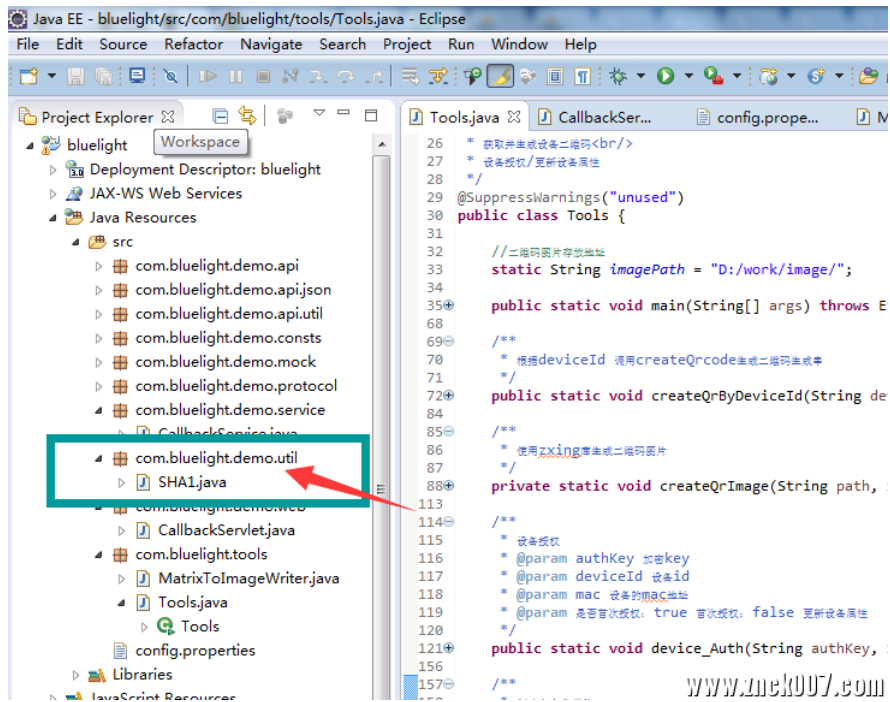


Tools.java共有main、createQrByDeviceId、createQrImage、device_Auth、createMenu等5个函数。
分别为主函数、建立二维码、生成图片、授权设备、建立菜单。
这里我们只跳用建立菜单，二维码和授权，之后在说硬件的时候再教大家。

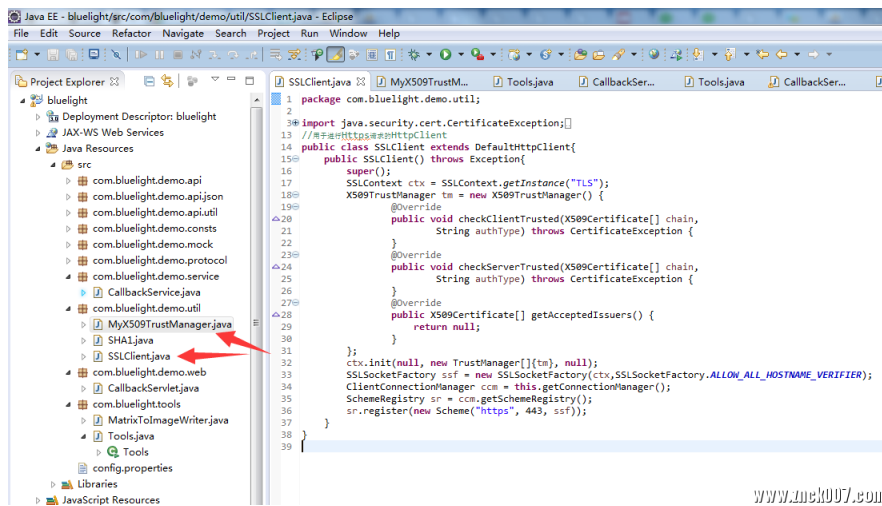
```
0
1
0
2 // 文本消息
0
3         if (MsgType.TEXT.equals(msgType)) {
0
4             // 可以在此处进行关键字自动回复
0
5             String content = "收到文本消息：" + reqMa
0
6
7             if (reqMap.get("Content").equals("你好")) {
0
8                 content = "你也好，你全家
0
9                 都好。";
0
10
11             }
0
12             if (reqMap.get("Content").equals("菜单") )
0
13
14                 Tools.createMenu();//建立
0
15             菜单
0
16             }
0
17             return XmlResp.buildText(fromUser, toUser,
0
18             content);
0
19
20         }
```

跟上篇一样，用户发送菜单二字，我们就调用并生成菜单
提醒：（菜单生成微信要返回主界面，再进行一次微信，同时设置第二次有时要24小时内才能生效）
但这里完成，经过小编测试，也还是不成功，原因是因为当前微信的demo不支持https请求（也不知道为什么微信demo自己不调试好），那我们再改改按下一个步骤来做就可以了。

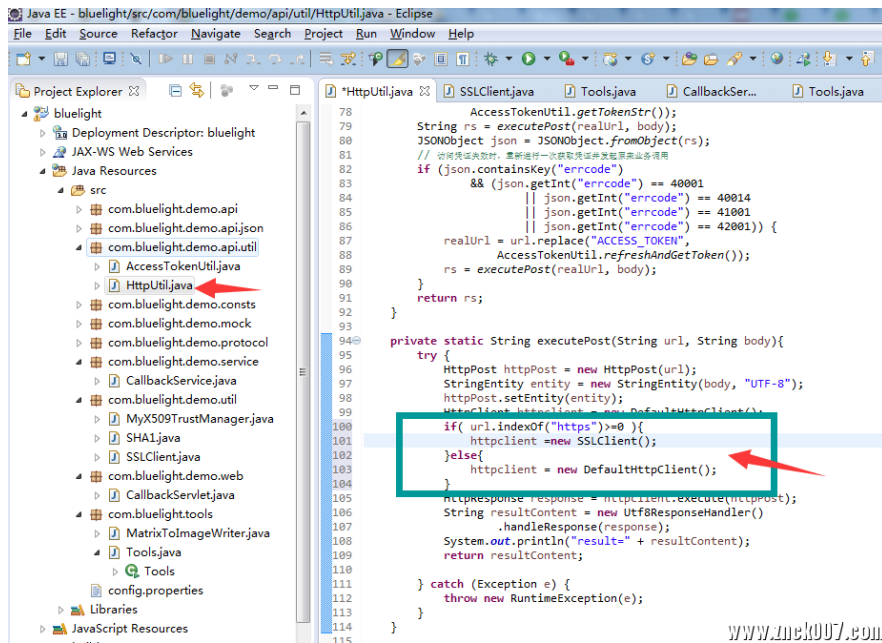
七、支持Https的POST（默认DEMO不支持，修改代码）



我们要在这里添加二个文件，原本是没有的。



添加进去，SSLClient.java、MyX509TrustManager.java这两个文件就是实现https的请求的（修改后的DEMO在最后面提供给大家下载的）



我们还需要在http请求的地方调用SSLClient类，如上图，我们在executeGet、executePost添加了一段代码。

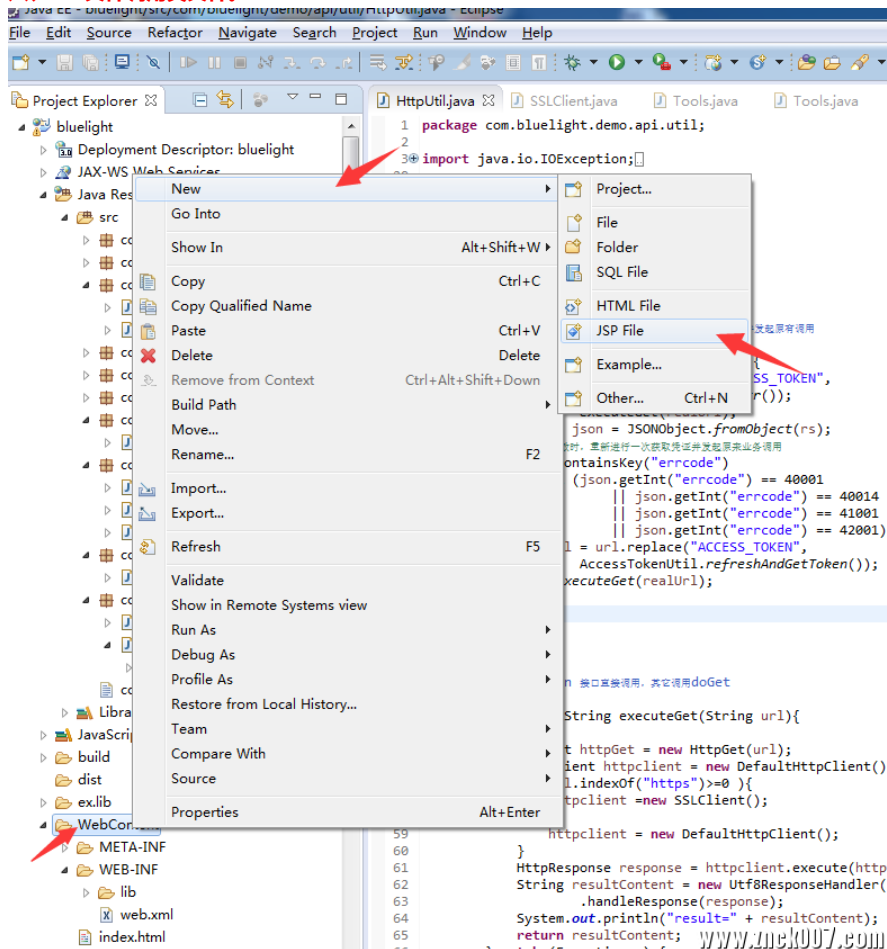
```

if( url.indexOf("https")>=0 ){
    httpClient =new SSLClient();
}else{
    httpClient = new DefaultHttpClient();
}

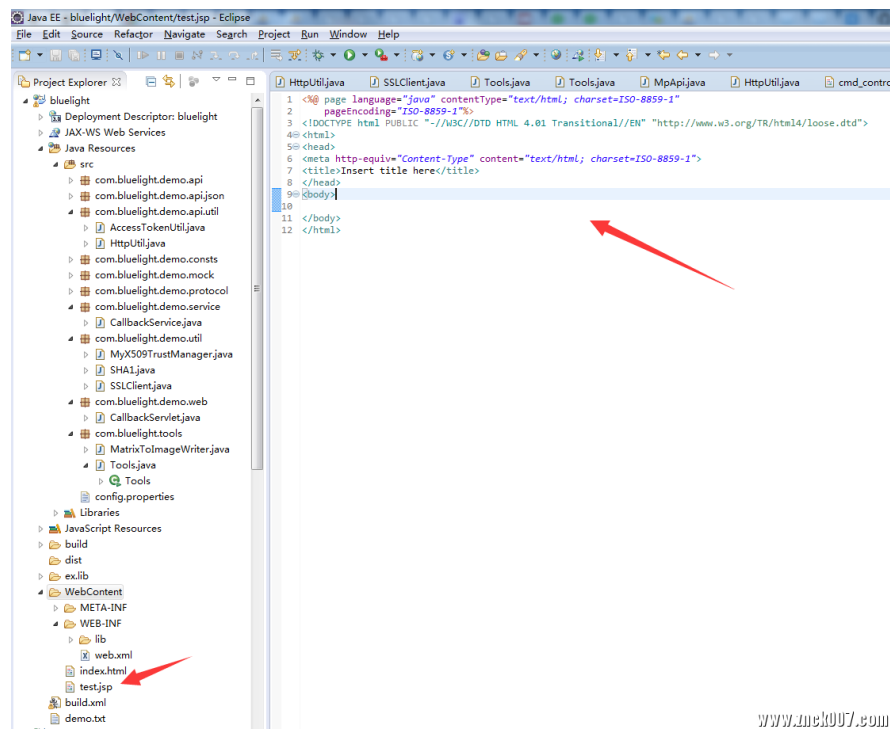
```

如果是https那么就自动请求ssl，实现可以请求https了。

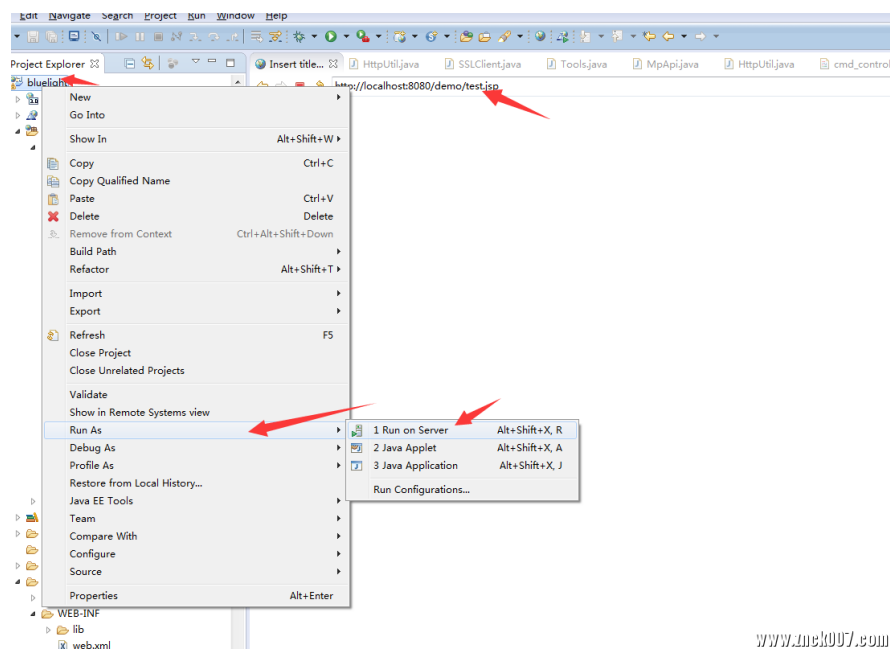
八、JSP文件调用类文件。



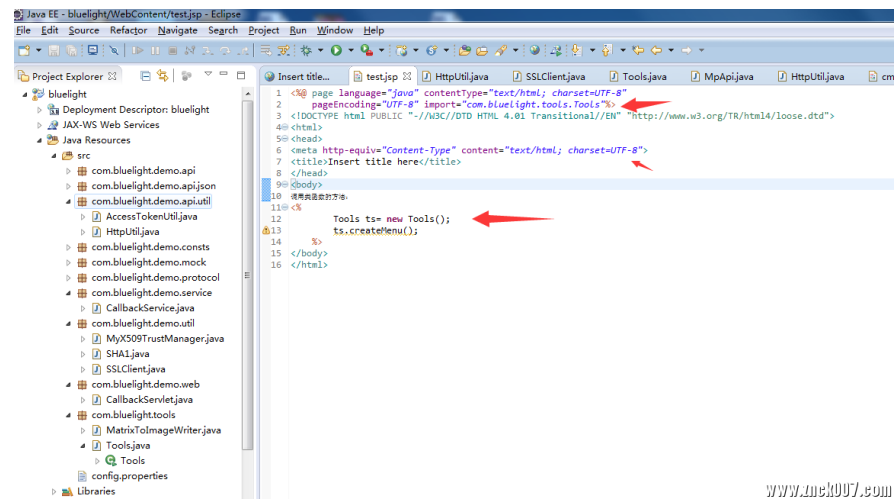
右击WebContent文件->New->JSP File，新建一个jsp文件，取名为test.jsp，然后点击finish。



自动生成jsp文件，然后运行。



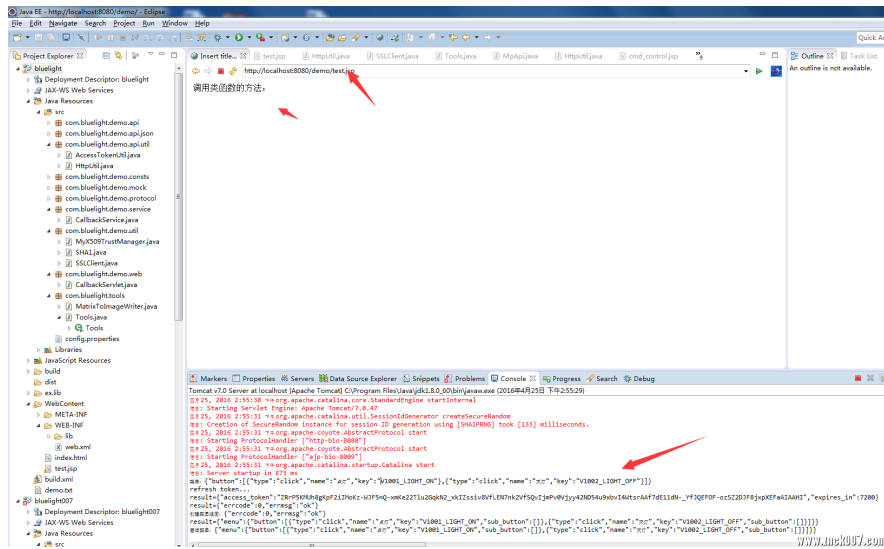
网址后面加test.jsp，然后回车，就能看到白白的页面，访问成功。



[Java] [纯文本查看](#) [复制代码](#)

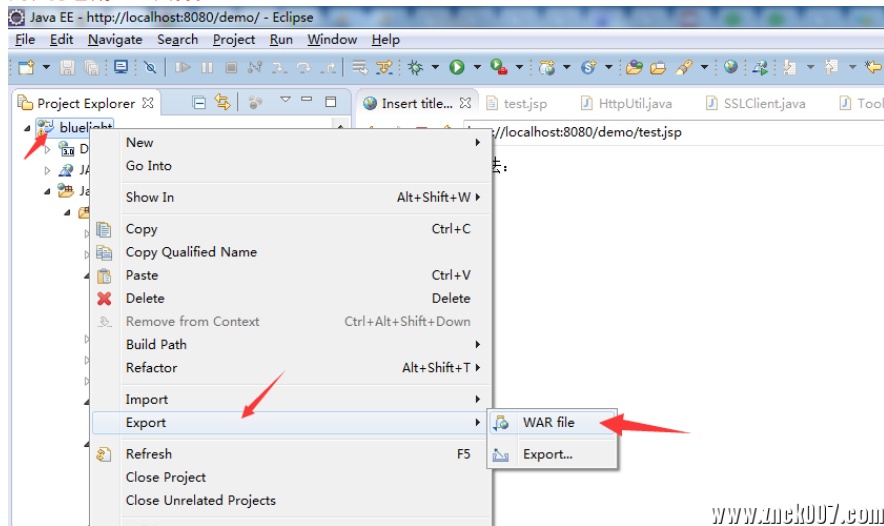
```
?
0
1
0
2
0
3
0
4 <%@ page language="java" contentType="text/html; charset=UTF-8"
0     pageEncoding="UTF-8" import="com.bluelight.tools.Tools"%>
5 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html
0 4/loose.dtd">
6 <html>
0 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
0 <title>Insert title here</title>
8 </head>
0 <body>
9 调用类函数的方法:
1 <%
0
1         Tools ts= new Tools();
1         ts.createMenu();
1     %>
2 </body>
1 </html>
3
1
4
1
5
1
6
```

这时再修改一个这个文件的代码如上图，这样就可以成功调用类文件了，但不建议这么调用不正规，只是方便而以。

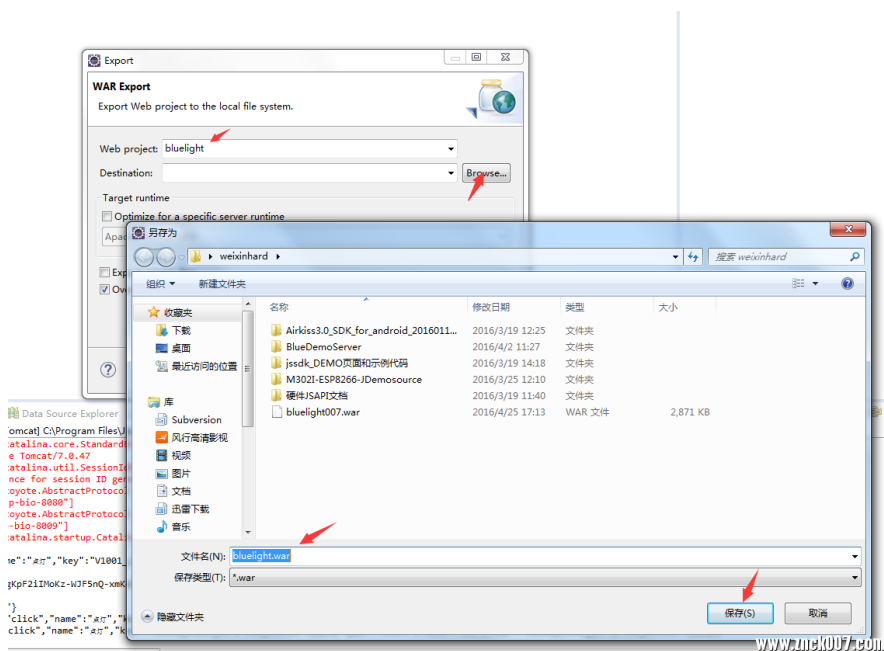


再运行，输出如下信息，说明成功！

九、打包成War文件。



开始导出war文件。



导出一个文件名，保存就能成功导出war文件，之后就可以上传到服务器了。

十、总结。

原来的DEMO不支持https请求，也没有jsp文件，这些都要自己建立的。
经过对微信Demo的源代码分析、修改、调式，那么我们这一课节就完成了。
附源代码下载：<http://pan.baidu.com/s/1skSC9jf>

请关注@智能创客 微信：znck007（打造DIY创客平台）

作者:mohongmao | 分类:微信硬件开发 | 浏览:1940 | 评论:0

[« 上一篇](#)

[下一篇 »](#)

发表评论:

<input type="text" value="访客"/>	名称(*)
<input type="text"/>	邮箱
<input type="text"/>	网址
<div>正文(*)</div> <div></div>	
<input type="button" value="提交"/>	

◎欢迎参与讨论，请在这里发表您的看法、交流您的观点。