

第5章



菜单和事件的应用开发

本章主要介绍自定义菜单和个性化菜单的要求、接口,以及如何通过这些接口实现自定义菜单。

5.1 说明

5.1.1 自定义菜单的要求

自定义菜单(使用普通自定义菜单创建接口创建的菜单称为默认菜单)能够丰富界面,让用户更好、更快地理解公众号的功能。自定义菜单最多包括 3 个一级菜单,每个一级菜单最多包含 5 个二级菜单。一级菜单最多 5 个汉字,二级菜单最多 8 个汉字,多出来的部分将会以“...”代替。

如果菜单有更新,就会刷新客户端的菜单。测试时可以尝试取消关注公众号后再次关注,则可以看到创建后的新菜单效果。

5.1.2 自定义菜单的按钮类型

自定义菜单接口可实现多种类型按钮。按钮类型和用户单击按钮后微信的反应如表 5-1 所示。

表 5-1 中第 4 行(scancode_push)到第 9 行(location_select)(第 1 列)的所有事件,仅支持微信 iPhone 5.4.1 以上版本和 Android 5.4 以上版本的微信用户。第 10 行和第 11 行(第 1 列)是专门给第三方平台旗下未微信认证的订阅号准备的事件类型,它们是没有事件推送的,能力相对受限。

表 5-1 按钮类型和用户单击按钮后微信的反应

类 型	用户单击按钮后微信的反应
click	通过消息接口推送消息类型为 event 的结构
view	打开在按钮中填写的网页 URL
scancode_push	将调起扫一扫工具,完成扫码操作后显示扫描结果(如果是 URL,将进入 URL)
scancode_waitmsg	将调起扫一扫工具,完成扫码操作后,回传扫码的结果,同时收起扫一扫工具,然后弹出“消息接收中”提示框
pic_sysphoto	将调起系统相机,完成拍照操作后,回传拍摄的相片,同时收起系统相机
pic_photo_or_album	文本消息将弹出选择器供用户选择“拍照”或者“从手机相册选择”
pic_weixin	将调起微信相册,完成选择操作后,将选择的相片发送给开发者的服务器,同时收起相册
location_select	将调起地理位置选择工具,完成选择操作后,将选择的地理位置发送给开发者的服务器,同时收起位置选择工具
media_id	将开发者填写的永久素材 id 下发给用户,草稿接口灰度完成后,用 article_id 代替它
view_limited	将打开在按钮中填写的永久素材 id 对应的图文消息 URL,草稿接口灰度完成后,用 article_view_limited 代替它

5.1.3 自定义菜单的接口

创建自定义菜单的接口 URL 为 https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN。

可以使用接口查询自定义菜单的结构。在设置个性化菜单后,使用查询接口可以获取默认菜单和全部个性化菜单信息。查询接口的 URL 为 https://api.weixin.qq.com/cgi-bin/get_current_selfmenu_info?access_token=ACCESS_TOKEN。

创建自定义菜单后,还可以删除当前使用的自定义菜单。在设置个性化菜单后,调用删除接口会删除默认菜单及全部个性化菜单。删除接口的 URL 为 https://api.weixin.qq.com/cgi-bin/menu/delete?access_token=ACCESS_TOKEN。

5.1.4 个性化菜单接口

为了帮助公众号实现灵活的业务运营,微信公众平台提供了个性化菜单接口,通过该接口,可以让公众号的不同用户群体看到不一样的自定义菜单。可以通过用户标签、用户性别、用户手机操作系统、用户客户端设置的地区和语言等条件来设置用户看到的菜单。

创建个性化菜单之前必须先创建默认菜单。个性化菜单的更新是会被覆盖的。例如公众号先后发布了默认菜单、个性化菜单 1、个性化菜单 2 和个性化菜单 3。那么当用户进入公众号页面时,将从个性化菜单 3 开始匹配,如果个性化菜单 3 匹配成功,则直接返回个性化菜单 3,否则继续尝试匹配个性化菜单 2,直到成功匹配到一个菜单。创建个性化菜单的接口 URL 为 https://api.weixin.qq.com/cgi-bin/menu/addconditional?access_token=ACCESS_TOKEN。

5.2 自定义菜单的应用开发

5.2.1 创建自定义菜单项类

继续在 4.3 节的基础上进行开发。在包 edu.bookcode 中创建 exofmenu 子包,并在包 edu.bookcode.exofmenu 中创建 menu 子包,在包 edu.bookcode.exofmenu.menu 中创建类 Button,代码如例 5-1 所示。

【例 5-1】 类 Button 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class Button {
    private String name;
}
```

在包 edu.bookcode.exofmenu.menu 中创建类 ClickButton,代码如例 5-2 所示。

【例 5-2】 类 ClickButton 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class ClickButton extends Button {
    private String type;
    private String key;
}
```

在包 edu.bookcode.exofmenu.menu 中创建类 ViewButton,代码如例 5-3 所示。

【例 5-3】 类 ViewButton 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class ViewButton extends Button {
    private String type;
    private String url;
}
```

在包 edu.bookcode.exofmenu.menu 中创建类 ScancodeButton,代码如例 5-4 所示。

【例 5-4】 类 ScancodeButton 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class ScancodeButton extends Button {
    private String type;
    private String key;
}
```



视频讲解

在包 edu.bookcode.exofmenu.menu 中创建类 PicButton,代码如例 5-5 所示。

【例 5-5】 类 PicButton 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class PicButton extends Button {
    private String type;
    private String key;
}
```

在包 edu.bookcode.exofmenu.menu 中创建类 LocationButton,代码如例 5-6 所示。

【例 5-6】 类 LocationButton 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class LocationButton extends Button {
    private String type;
    private String key;
}
```

在包 edu.bookcode.exofmenu.menu 中创建类 ComplexButton,代码如例 5-7 所示。

【例 5-7】 类 ComplexButton 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class ComplexButton extends Button {
    private Button[] sub_button;
}
```

在包 edu.bookcode.exofmenu.menu 中创建类 Menu,代码如例 5-8 所示。

【例 5-8】 类 Menu 的代码示例。

```
package edu.bookcode.exofmenu.menu;
import lombok.Data;
@Data
public class Menu {
    private Button[] button;
}
```

5.2.2 创建类 TextMessageToXML

在包 edu.bookcode.exofmenu 中创建 util 子包,在包 edu.bookcode.exofmenu.util 中创建类 TextMessageToXML,代码如例 5-9 所示。

【例 5-9】 类 TextMessageToXML 的代码示例。

```
package edu.bookcode.exofmenu.util;
//导入前面的类,也可以将前面的类复制到包 edu.bookcode.exofmenu 中
```

```
import edu.bookcode.exofmessage.message.resp.Article;
import edu.bookcode.exofmessage.message.resp.NewsMessage;
import edu.bookcode.exofmessage.message.resp.TextMessage;
import java.util.Date;
import java.util.List;
import java.util.Map;
public class TextMessageToXML {
    public static String messageToXML(Map<String, String> message, String content){
        TextMessage textMessage = new TextMessage();
        textMessage.setToUserName(message.get("ToUserName"));
        textMessage.setFromUserName(message.get("FromUserName"));
        textMessage.setCreateTime(new Date().getTime());
        textMessage.setContent(content);
        textMessage.setMsgType("text");
        String xml = textMessageToXML(textMessage);
        return xml;
    }
    private static String textMessageToXML(TextMessage textMessage) {
        String xml = "<xml>" +
            "<ToUserName>" + textMessage.getFromUserName() + "</ToUserName>" +
            "<FromUserName>" + textMessage.getToUserName() + "</FromUserName>" +
            "<CreateTime>" + textMessage.getCreateTime() + "</CreateTime>" +
            "<MsgType>text</MsgType>" +
            "<Content>" + textMessage.getContent() + "</Content>" +
            "</xml>";
        return xml;
    }
    public static String newsToXML(Map<String, String> message, List<Article> articleList )
    {
        NewsMessage newsMessage = new NewsMessage();
        newsMessage.setToUserName(message.get("ToUserName"));
        newsMessage.setFromUserName(message.get("FromUserName"));
        newsMessage.setCreateTime(new Date().getTime());
        newsMessage.setMsgType("news");
        newsMessage.setArticleCount(articleList.size());
        newsMessage.setArticles(articleList);
        String xml = newsMessageToXML(newsMessage);
        return xml;
    }
    private static String newsMessageToXML(NewsMessage newsMessage) {
        String xml = "<xml>" +
            "<ToUserName>" + newsMessage.getFromUserName() + "</ToUserName>" +
            "<FromUserName>" + newsMessage.getToUserName() + "</FromUserName>" +
            "<CreateTime>" + newsMessage.getCreateTime() + "</CreateTime>" +
            "<MsgType>news</MsgType>" +
            "<ArticleCount>1</ArticleCount>" +
            "<Articles>" +
            "<item>" +
            "<Title>" + newsMessage.getArticles().get(0).getTitle() + "</Title>" +
            "<Description>" + newsMessage.getArticles().get(0).getDescription() +
            "</Description>" +
```

```

        "        <PicUrl>" + newsMessage.getArticles().get(0).getPicUrl() +
        "    </PicUrl>" +
        "        <Url>" + newsMessage.getArticles().get(0).getUrl() + "</Url>" +
        "    </item>" +
        " </Articles>" +
        "</xml>";

    return xml;
}

public static String processScanPush(String fromUserName, String toUserName) {
    String xml = "<xml>" +
        "<ToUserName>" + fromUserName + "</ToUserName>" +
        "<FromUserName>" + toUserName + "</FromUserName>" +
        "<CreateTime>" + new Date().getTime() + "</CreateTime>" +
        "<MsgType> event </MsgType>" +
        "<Event> scancode_push </Event>" +
        "<EventKey> rselfmenu22 </EventKey>" +
        "<ScanCodeInfo>" +
        "<ScanType> qrcode </ScanType>" +
        "<ScanResult> 1 </ScanResult>" +
        "</ScanCodeInfo>" +
        "</xml>";

    return xml;
}

public static String processScanWaitMsg(String fromUserName, String toUserName) {
    String xml = "<xml>" +
        "<ToUserName>" + fromUserName + "</ToUserName>" +
        "<FromUserName>" + toUserName + "</FromUserName>" +
        "<CreateTime>" + new Date().getTime() + "</CreateTime>" +
        "<MsgType> event </MsgType>" +
        "<Event> scancode_waitmsg </Event>" +
        "<EventKey> rselfmenu21 </EventKey>" +
        "<ScanCodeInfo>" +
        "<ScanType> qrcode </ScanType>" +
        "<ScanResult> 1 </ScanResult>" +
        "</ScanCodeInfo>" +
        "</xml>";

    return xml;
}

public static String processSysphoto(String fromUserName, String toUserName) {
    String xml = "<xml>" +
        "<ToUserName>" + fromUserName + "</ToUserName>" +
        "<FromUserName>" + toUserName + "</FromUserName>" +
        "<CreateTime>" + new Date().getTime() + "</CreateTime>" +
        "<MsgType> event </MsgType>" +
        "<Event> pic_sysphoto </Event>" +
        "<EventKey> 6 </EventKey>" +
        "<SendPicsInfo><Count> 1 </Count>" +
        "<PicList><item><PicMd5Sum> 1b5f7c23b5bf75682a53e7b6d163e185" +
        "</PicMd5Sum>\n" +
        "</item>\n" +
        "</PicList>\n" +

```

```

        "</SendPicsInfo>" +
        "</xml>";
    return xml;
}

public static String processPhotoOrAlbum(String fromUserName, String toUserName) {
    String xml = "< xml>" +
        "< ToUserName>" + fromUserName + "</ToUserName>" +
        "< FromUserName>" + toUserName + "</FromUserName>" +
        "< CreateTime>" + new Date().getTime() + "</CreateTime>" +
        "< MsgType> event</MsgType>" +
        "< Event> pic_photo_or_album</Event>" +
        "< EventKey> rselfmenu24</EventKey>" +
        "< SendPicsInfo>< Count> 1</Count>" +
        "< PicList>< item>< PicMd5Sum> 5a75aaca956d97be686719218f275c6b" +
        "</PicMd5Sum>\n" +
        "</item>\n" +
        "</PicList>\n" +
        "</SendPicsInfo>" +
        "</xml>";

    return xml;
}

public static String processPicWeiXin(String fromUserName, String toUserName) {
    String xml = "< xml>" +
        "< ToUserName>" + fromUserName + "</ToUserName>" +
        "< FromUserName>" + toUserName + "</FromUserName>" +
        "< CreateTime>" + new Date().getTime() + "</CreateTime>" +
        "< MsgType> event</MsgType>" +
        "< Event> pic_weixin</Event>" +
        "< EventKey> rselfmenu25</EventKey>" +
        "< SendPicsInfo>< Count> 1</Count>" +
        "< PicList>< item>< PicMd5Sum> 5a75aaca956d97be686719218f275c6b" +
        "</PicMd5Sum>\n" +
        "</item>\n" +
        "</PicList>\n" +
        "</SendPicsInfo>" +
        "</xml>";

    return xml;
}

public static String processLocation(String fromUserName, String toUserName) {
    String xml = "< xml>" +
        "< ToUserName>" + fromUserName + "</ToUserName>" +
        "< FromUserName>" + toUserName + "</FromUserName>" +
        "< CreateTime>" + new Date().getTime() + "</CreateTime>" +
        "< MsgType> event</MsgType>" +
        "< Event> location_select</Event>" +
        "< EventKey> rselfmenu26</EventKey>" +
        "< SendLocationInfo>< Location_X> 23></Location_X>\n" +
        "< Location_Y> 113</Location_Y>\n" +
        "< Scale> 15</Scale>\n" +
        "< Label>广州市海珠区客村艺苑路 106 号</Label>\n" +
        "< Poiname></Poiname>\n" +

```

```

        "</SendLocationInfo>" +
        "</xml>";
    return xml;
}
}

```

5.2.3 创建类 MenuUtil

在包 edu.bookcode.exofmenu.util 中创建类 MenuUtil, 代码如例 5-10 所示。

【例 5-10】 类 MenuUtil 的代码示例。

```

package edu.bookcode.exofmenu.util;
import edu.bookcode.exofmenu.menu.Menu;
import edu.bookcode.service.CommonUtil; //导入
import net.sf.json.JSONObject;
public class MenuUtil {
//创建、查询、删除相关 API 的 URL
public final static String menu_create_url = "https://api.weixin.qq.com/cgi-bin/menu/
create?access_token=ACCESS_TOKEN";
public final static String menu_get_url = "https://api.weixin.qq.com/cgi-bin/menu/get?
access_token=ACCESS_TOKEN";
public final static String menu_delete_url = "https://api.weixin.qq.com/cgi-bin/menu/
delete?access_token=ACCESS_TOKEN";
//创建自定义菜单
public static boolean createMenu(Menu menu, String accessToken) {
    boolean result = false;
    String url = menu_create_url.replace("ACCESS_TOKEN", accessToken);
    String jsonMenu = JSONObject.fromObject(menu).toString();
    JSONObject jsonObject = CommonUtil.httpsRequest(url, "POST", jsonMenu);
    if (null != jsonObject) {
        int errorCode = jsonObject.getInt("errcode");
        if (0 == errorCode) {
            result = true;
        } else {
            result = false;
        }
        System.out.println("errcode:{" + jsonObject.getInt("errcode") + "}, errmsg:{" + jsonObject.
getString("errmsg") + "}"");
    }
    }
    return result;
}
//查询菜单信息
public static String getMenu(String accessToken) {
    String result = null;
    String requestUrl = menu_get_url.replace("ACCESS_TOKEN", accessToken);
    JSONObject jsonObject = CommonUtil.httpsRequest(requestUrl, "GET", null);
    if (null != jsonObject) {
        result = jsonObject.toString();
    }
}

```

```

        return result;
    }
    //删除菜单
    public static boolean deleteMenu(String accessToken) {
        boolean result = false;
        String requestUrl = menu_delete_url.replace("ACCESS_TOKEN", accessToken);
        JSONObject jsonObject = CommonUtil.httpsRequest(requestUrl, "GET", null);
        if (null != jsonObject) {
            int errorCode = jsonObject.getInt("errcode");
            if (0 == errorCode) {
                result = true;
            } else {
                result = false;
            }
        }
        System.out.println("errcode:{" + jsonObject.getInt("errcode") + "},errmsg:{" + jsonObject.getString("errmsg") + "}");
    }
    return result;
}
}

```

5.2.4 创建类 ButtonMenuService

在包 edu.bookcode.exofmenu 中创建 service 子包,在包 edu.bookcode.exofmenu.service 中创建类 ButtonMenuService,代码如例 5-11 所示。

【例 5-11】 类 ButtonMenuService 的代码示例。

```

package edu.bookcode.exofmenu.service;
import java.util.*;
import javax.servlet.http.HttpServletRequest;
import edu.bookcode.exofmenu.util.TextMessageToXML;
//导入前面的类,也可以将前面的类复制到包 edu.bookcode.exofmenu 中
import edu.bookcode.exofmessage.message.resp.Article;
import edu.bookcode.exofmessage.util.MessageUtil;
public class ButtonMenuService {
    public static String processRequest(HttpServletRequest request) {
        String xml;
        try {
            Map<String, String> requestMap = MessageUtil.parseXml(request);
            String msgType = requestMap.get("MsgType");
            String fromUserName = requestMap.get("FromUserName");
            String toUserName = requestMap.get("ToUserName");
            String content = "";
            String eventKey;
            if (msgType.equals("event")) {
                String eventType = requestMap.get("Event").toLowerCase();
                switch (eventType) {
                    case "subscribe":
                        content = "感谢您的关注!";

```

```

        break;
    case "unsubscribe":
        //取消关注后,用户不会再收到公众号发送的消息,因此不需要回复
        break;
        //扫码后会开始接收推送消息(官方文档中将其简称为扫码堆事件)的
        //事件推送
    case "scancode_push":
        xml = TextMessageToXML.processScanPush(fromUserName,toUserName);
        return xml;
        //扫码带提示的事件推送
    case "scancode_waitmsg":
        xml = TextMessageToXML.processScanWaitMsg(fromUserName,toUserName);
        return xml;
    case "pic_sysphoto":
        xml = TextMessageToXML.processSysphoto(fromUserName,toUserName);
        return xml;
    case "pic_photo_or_album":
        xml = TextMessageToXML.processPhotoOrAlbum(fromUserName,toUserName);
        return xml;
    case "pic_weixin":
        xml = TextMessageToXML.processPicWeiXin(fromUserName,toUserName);
        return xml;
        //此例省略了 location_select 等情形,读者可以参考完成
    case "click":
        eventKey = requestMap.get("EventKey");
        //根据 key 值判断用户单击的按钮
        if (eventKey.equals("QQ")) {
            Article article = new Article();
            article.setTitle("QQ 的联系方式");
            article.setDescription("QQ 不一定能及时回复.");
            article.setPicUrl("");
            article.setUrl("http://qq.com");
            List<Article> articleList = new ArrayList<Article>();
            articleList.add(article);
            //创建图文消息
            xml = TextMessageToXML.newsToXML(requestMap,articleList);
            return xml;
        } else if (eventKey.equals("WeiXin")) {
            content = "微信号:jsnuws";
        } else if (eventKey.equals("Phone")) {
            content = "手机号:12345678901";
        } else if (eventKey.equals("Email")) {
            content = "邮箱:6780912345@qq.com";
        }
        break;
    default:
        break;
    }
}

xml = TextMessageToXML.messageToXML(requestMap,content);
return xml;

```

```
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return "error";  
    }  
}
```

5.2.5 创建类 MenuInit

在包 edu.bookcode.exofmenu 中创建类 MenuInit, 该类主要定义了菜单信息, 代码如例 5-12 所示。

【例 5-12】 类 MenuInit 的代码示例。

```
package edu.bookcode.exofmenu;  
import edu.bookcode.exofmenu.menu.*;  
import edu.bookcode.exofmenu.util.MenuUtil;  
import edu.bookcode.service.TemptTokenUtil;  
public class MenuInit {  
    //注意菜单层级、子项、命名等规定的限制  
    public static Menu getMenu() {  
        //第 1 列子菜单中第 1 项子菜单项  
        ViewButton btn11 = new ViewButton();  
        btn11.setName("微信小程序开发基础");  
        btn11.setType("view");  
        btn11.setUrl("https://item.jd.com/10026528815782.html");  
        //第 1 列子菜单中第 2 项子菜单项  
        ViewButton btn12 = new ViewButton();  
        btn12.setName("微信小程序云开发");  
        btn12.setType("view");  
        btn12.setUrl("https://item.jd.com/12958844.html");  
        ViewButton btn13 = new ViewButton();  
        btn13.setName("Spring Boot 区块链应用开发入门");  
        btn13.setType("view");  
        btn13.setUrl("https://item.jd.com/12735489.html");  
        ViewButton btn14 = new ViewButton();  
        btn14.setName("Spring Boot 开发实战");  
        btn14.setType("view");  
        btn14.setUrl("https://item.jd.com/10026542588356.html");  
        ViewButton btn15 = new ViewButton();  
        btn15.setName("Spring Cloud 微服务开发实战");  
        btn15.setType("view");  
        btn15.setUrl("https://item.jd.com/10026550550811.html");  
        ScancodeButton btn21 = new ScancodeButton();  
        btn21.setName("扫码带提示");  
        btn21.setType("scancode_waitmsg");  
        btn21.setKey("rselfmenu21");  
        ScancodeButton btn22 = new ScancodeButton();  
        btn22.setName("扫码推事件");  
        btn22.setType("scancode_push");
```

```

        btn22.setKey("rselfmenu22");
        PicButton btn23 = new PicButton();
        btn23.setName("系统拍照发图");
        btn23.setType("pic_sysphoto");
        btn23.setKey("rselfmenu23");
        PicButton btn24 = new PicButton();
        btn24.setName("拍照或者相册发图");
        btn24.setType("pic_photo_or_album");
        btn24.setKey("rselfmenu24");
        PicButton btn25 = new PicButton();
        btn25.setName("微信相册发图");
        btn25.setType("pic_weixin");
        btn25.setKey("rselfmenu25");
        ClickButton btn31 = new ClickButton();
        btn31.setName("QQ");
        btn31.setType("click");
        btn31.setKey("QQ");
        ClickButton btn32 = new ClickButton();
        btn32.setName("WeiXin");
        btn32.setType("click");
        btn32.setKey("WeiXin");
        ClickButton btn33 = new ClickButton();
        btn33.setName("Phone");
        btn33.setType("click");
        btn33.setKey("Phone");
        ClickButton btn34 = new ClickButton();
        btn34.setName("Email");
        btn34.setType("click");
        btn34.setKey("Email");
        //第1列子菜单
        ComplexButton mainBtn1 = new ComplexButton();
        mainBtn1.setName("图书");
        mainBtn1.setSub_button(new Button[] { btn11, btn12, btn13, btn14, btn15 });
        ComplexButton mainBtn2 = new ComplexButton();
        mainBtn2.setName("扫码和发图");
        mainBtn2.setSub_button(new Button[] { btn21, btn22, btn23, btn24, btn25 });
        ComplexButton mainBtn3 = new ComplexButton();
        mainBtn3.setName("联系方式");
        mainBtn3.setSub_button(new Button[] { btn31, btn32, btn33, btn34 });
        Menu menu = new Menu();
        menu.setButton(new Button[] { mainBtn1, mainBtn2, mainBtn3 });
        return menu;
    }
    //使用 main 方法是为了简化测试的需要,实际开发时可以自动创建菜单
    public static void main(String[] args) {
        boolean result = MenuUtil.createMenu(getMenu(), new TemptTokenUtil().getTokenInfo());
        if (result)
            System.out.println("菜单创建成功!");
        else
            System.out.println("菜单创建失败!");
    }
}

```

5.2.6 创建类 ExOfMenuController

在包 edu.bookcode.exofmenu 中创建 controller 子包,在包 edu.bookcode.exofmenu.controller 中创建类 ExOfMenuController,代码如例 5-13 所示。

【例 5-13】 类 ExOfMenuController 的代码示例。

```
package edu.bookcode.exofmenu.controller;
import edu.bookcode.exofmenu.MenuInit;
import edu.bookcode.exofmenu.service.ButtonMenuService;
import edu.bookcode.exofmenu.util.MenuUtil;
//导入前面的类
import edu.bookcode.service.TemptTokenUtil;
import edu.bookcode.util.OutAndSendUtil;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@RestController
public class ExOfMenuController {
    //下面一行是运行本类时的相对地址
    @RequestMapping("/")
    //为了测试方便,在运行其他类时,必须注释掉上一行代码,即修改相对地址
    //并可以去掉下一行代码的注释,修改本类的相对地址
    //@RequestMapping("/testMenu")
    public void testMenu (HttpServletRequest request, HttpServletResponse response) throws
    ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setCharacterEncoding("UTF-8");
        String respXml = ButtonMenuService.processRequest(request);
        String xml = "";
        if(respXml.contains("< Content ></Content >")) {
            xml = respXml.replace("< Content ></Content >","< Content >您好,有什么可以帮到您?</Content >");
            System.out.println(xml);
        } else {
            xml = respXml;
            System.out.println(xml);
        }
        OutAndSendUtil.sendMessageToWXAppClient(xml,response);
        //首次注释下一行,运行程序
        //menuProcess();
        //第 2 次、第 3 次取消注释,再运行程序
        //第 4 次取消下面两行注释
        //System.out.println("重新增加菜单后:" + new MyMenuExDemo().getSpecialMenuJson("2",
        "Java"));
    }
    //对菜单的创建、查询和删除
```

```
private void menuProcess() {  
    String accessToken = new TemptTokenUtil().getTokenInfo();  
    //显示菜单信息  
    System.out.println("菜单信息:" + MenuUtil.getMenu(accessToken));  
    //显示菜单删除,菜单为空  
    System.out.println("删除菜单后:" + MenuUtil.deleteMenu(accessToken));  
    //第2次运行时,注释掉下一行的语句,可以观察到删除菜单后菜单为空  
    System.out.println("重新增加菜单后:" + MenuUtil.createMenu(MenuInit.getMenu(), accessToken));  
    //第3次运行时,启用上一行的语句,可以观察到菜单再次出现  
}  
}
```

5.2.7 运行程序

启动内网穿透工具后,按照例 4-46 中注释给出的提示修改 ExOfMessageController 的相对地址,并在 IDEA 中先运行类 MenuInit 再运行项目入口类 WxgzptkfbbookApplication。

手机微信公众号中第 1 级菜单如图 5-1 所示,第 2 级菜单第 1 列如图 5-2 所示,第 2 级菜单第 2 列如图 5-3 所示,第 2 级菜单第 3 列如图 5-4 所示。单击图 5-2 中“Spring Cloud 微服务开发实战”菜单项,跳转到对应网址的图书页面,如图 5-5 所示。单击图 5-3 中“拍照或者相册发图”菜单项,结果如图 5-6 所示。依次单击图 5-4 中 QQ、WeiXin 菜单项,结果如图 5-7 所示。在图 5-6 中选择“拍照”或“从相册选择”后发送图片,结果如图 5-8 所示。



图 5-1 第 1 级菜单在手机微信公众号中的输出(底部)



图 5-2 第 2 级菜单第 1 列(图书)在手机微信公众号中的输出



图 5-3 第 2 级菜单第 2 列(扫码和发图)在手机微信公众号中的输出



图 5-4 第 2 级菜单第 3 列(联系方式)在手机微信公众号中的输出



图 5-5 单击图 5-2 中“Spring Cloud 微服务开发实战”菜单项后跳转到对应网址的图书页面

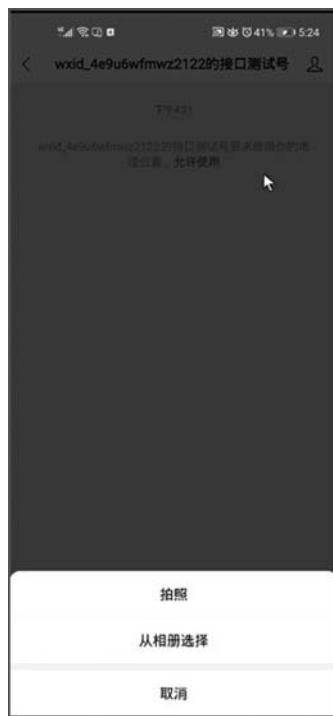


图 5-6 单击图 5-3 中“拍照或者相册发图”菜单项的结果



图 5-7 依次单击图 5-4 中 QQ、WeiXin 菜单项的结果



图 5-8 在图 5-6 中选择“拍照”或“从相册选择”后发送图片的结果(部分)

按照例 5-13 所示修改类 ExOfMenuController 的代码(修改注释,详细操作可以参考视频讲解),可以得到菜单信息在控制台的输出结果如图 5-9 所示,删除菜单之后控制台的输出结果如图 5-10 所示,增加菜单之后控制台的输出结果如图 5-11 所示。单击图 5-4 中 WeiXin 菜单项后控制台的输出结果如图 5-12 所示。

```
菜单信息: {"menu":{"button":[{"name":"图书","sub_button":[{"name":"微信小程序开发基础","sub_button":[,
"type":"view","url":"https://item.jd.com/10026528815782.html"}, {"name":"微信小程序云开发",
"sub_button":[, "type":"view","url":"https://item.jd.com/12958844.html"}, {"name":"Spring Boot区块链应
用开发入门","sub_button":[, "type":"view","url":"https://item.jd.com/12735489.html"}, {"name":"Spring
Boot开发实战","sub_button":[, "type":"view","url":"https://item.jd.com/10026542588356.html"},
{"name":"Spring Cloud 微服务开发实战","sub_button":[, "type":"view","url":"https://item.jd.
com/10026550550811.html"}]}, {"name":"扫码和发图","sub_button":[{"name":"扫码带提示","sub_button":[,
"type":"scancode_waitmsg","key":"rselfmenu21"}, {"name":"扫码推事件","sub_button":[,
"type":"scancode_push","key":"rselfmenu22"}, {"name":"系统拍照发图","sub_button":[,
"type":"pic_sysphoto","key":"rselfmenu23"}, {"name":"拍照或者相册发图","sub_button":[,
"type":"pic_photo_or_album","key":"rselfmenu24"}, {"name":"微信相册发图","sub_button":[,
"type":"pic_weixin","key":"rselfmenu25"}]}, {"name":"联系方式","sub_button":[{"name":"QQ",
"sub_button":[, "type":"click","key":"QQ"}, {"name":"WeiXin","sub_button":[, "type":"click",
"key":"WeiXin"}, {"name":"Phone","sub_button":[, "type":"click","key":"Phone"}, {"name":"Email",
"sub_button":[, "type":"click","key":"Email"}]}]}}
```

图 5-9 菜单信息在控制台的输出结果(部分)

```
菜单信息: {"errcode":46003,"errmsg":"menu no exist rid: 60c46c9c-4036bb87-13b1c2a8"}  
删除菜单后: true
```

图 5-10 删除菜单之后控制台的输出结果(部分)

```
菜单信息: {"errcode":46003,"errmsg":"menu no exist rid: 60c46e50-33550127-766e5533"}  
删除菜单后: true  
重新增加菜单后: true
```

图 5-11 增加菜单之后控制台的输出结果(部分)

```
<xml><ToUserName>obKWL6Q6awrcWSKz3LeSmc0Yubfc</ToUserName><FromUserName>gh_0acb8bcc8eef</FromUserName>  
><CreateTime>1623491220734</CreateTime><MsgType>text</MsgType><Content>微信号: jsnuws</Content></xml>  
菜单信息: {"errcode":46003,"errmsg":"menu no exist rid: 60c48296-38ebbf82-22201100"}  
删除菜单后: true  
重新增加菜单后: true
```

图 5-12 单击图 5-4 中 WeiXin 菜单项后控制台的输出结果

习题 5

简答题

1. 简述对自定义菜单要求的理解。
2. 简述对自定义菜单按钮类型的理解。

实验题

1. 实现示例: 自定义菜单的应用开发。
2. 用自定义菜单独立完成一个实例。