## 微信SDK使用说明

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修订日期 | 修订者 | 修订内容 |
| V1.0 | 2014-03-12 | Caspar | 发布版本。 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. 引入SDK

将WechatSDK.jar加入到项目中

1. 引入配置文件

log4j.properties

wechat.properties

wechat.properties参数详解

|  |  |
| --- | --- |
| 参数名 | 参数意思 |
| token | Token, 与接口配置信息中的Token要一致 |
| appid | 第三方用户唯一凭证 |
| appSecret | 第三方用户唯一凭证密钥 |

1. 项目中需要的jar包

commons-beanutils-1.7.0.jar

commons-collections-3.2.jar

commons-lang-2.4.jar

commons-logging-1.1.1.jar

dom4j-1.6.1.jar

ezmorph-1.0.4.jar

json-lib-2.2.2-jdk15.jar

jsoup-1.7.2.jar

log4j-1.2.17.jar

xstream-1.1.2.jar

1. 建立与微信交互的接口

Example：

|  |
| --- |
| /\*\*  \* 处理微信请求servlet  \* **@author** caspar.chen  \* **@version** 1.0  \*  \*/  **public** **class** WeChatServlet **extends** HttpServlet {  **private** **static** **final** **long** *serialVersionUID* = 1L;  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  // 随机字符串  String echostr = request.getParameter("echostr");  PrintWriter out = **null**;  **try** {  out = response.getWriter();  // 通过检验signature对请求进行校验，若校验成功则原样返回echostr，否则接入失败  **if** (SignService.*checkSignature*(request)) {  out.print(echostr);  }  } **catch** (IOException e) {  e.printStackTrace();  } **finally** {  out.close();  out = **null**;  }  }  **public** **void** doPost(HttpServletRequest request, HttpServletResponse response)  **throws** ServletException, IOException {  **try** {  request.setCharacterEncoding("UTF-8");  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  response.setCharacterEncoding("UTF-8");  // 调用核心业务类接收消息、处理消息  String respMessage = CoreService.*processWebchatRequest*(request);  // 响应消息  PrintWriter out = **null**;  **try** {  out = response.getWriter();  out.print(respMessage);  } **catch** (IOException e) {  e.printStackTrace();  } **finally** {  out.close();  out = **null**;  }  }  } |

1. 建立处理消息的service

Example：

|  |
| --- |
| /\*\*  \* 处理微信发来的请求  \*  \* **@param** request  \* **@return** String  \*/  **public** **static** String processWebchatRequest(HttpServletRequest request) {  String respMessage = **null**;  **try** {  // xml请求解析  Map<String, String> requestMap = MessageUtil.*parseXml*(request);  // 消息类型  String msgType = requestMap.get("MsgType");  TextMessage textMessage = (TextMessage) MessageService  .*bulidBaseMessage*(requestMap,  ConstantWeChat.*RESP\_MESSAGE\_TYPE\_TEXT*);  NewsMessage newsMessage = (NewsMessage) MessageService  .*bulidBaseMessage*(requestMap,  ConstantWeChat.*RESP\_MESSAGE\_TYPE\_NEWS*);  String respContent = "";  // 文本消息  **if** (msgType.equals(ConstantWeChat.*REQ\_MESSAGE\_TYPE\_TEXT*)) {  // 接收用户发送的文本消息内容  String content = requestMap.get("Content");  List<Article> articleList = **new** ArrayList<Article>();  // 单图文消息  **if** ("1".equals(content)) {  Article article = **new** Article();  article.setTitle("测试TITLE");  article.setDescription("测试Description，长一点，测试Description，长一点，测试Description，长一点，测试Description，长一点");  article.setPicUrl("http://www.isicchina.com/images/apply\_now.jpg");  article.setUrl(RequestUtil.*getBasePath*(request)  + "view.html?id=1");  articleList.add(article);  // 设置图文消息个数  newsMessage.setArticleCount(articleList.size());  // 设置图文消息包含的图文集合  newsMessage.setArticles(articleList);  // 将图文消息对象转换成xml字符串  respMessage = MessageService.*bulidSendMessage*(newsMessage,  ConstantWeChat.*RESP\_MESSAGE\_TYPE\_NEWS*);  } **else** **if** ("#".equals(content)) {  textMessage.setContent("###");  respMessage = MessageService.*bulidSendMessage*(textMessage,  ConstantWeChat.*RESP\_MESSAGE\_TYPE\_TEXT*);  }  } **else** **if** (msgType.equals(ConstantWeChat.*REQ\_MESSAGE\_TYPE\_EVENT*)) {  // 事件类型  String eventType = requestMap.get("Event");  **if** (eventType.equals(ConstantWeChat.*EVENT\_TYPE\_SUBSCRIBE*)) {  // 关注  respContent = "感谢您关注偶,这里会给您提供最新的公司资讯和公告！\n";  } **else** **if** (eventType  .equals(ConstantWeChat.*EVENT\_TYPE\_UNSUBSCRIBE*)) {  // 取消关注,用户接受不到我们发送的消息了，可以在这里记录用户取消关注的日志信息  } **else** **if** (eventType.equals(ConstantWeChat.*EVENT\_TYPE\_CLICK*)) {  // 事件KEY值，与创建自定义菜单时指定的KEY值对应  String eventKey = requestMap.get("EventKey");  // 自定义菜单点击事件  **if** (eventKey.equals("11")) {  respContent = "天气预报菜单项被点击！";  } **else** **if** (eventKey.equals("12")) {  respContent = "公交查询菜单项被点击！";  }  }  textMessage.setContent(respContent);  respMessage = MessageService.*bulidSendMessage*(textMessage,  ConstantWeChat.*RESP\_MESSAGE\_TYPE\_TEXT*);  }  } **catch** (Exception e) {  e.printStackTrace();  }  **return** respMessage;  } |