# **联通chatbot说明文档**

## **一、关于手机**

手机要求：普通安卓手机且下载了中讯的短信APP

搜索chatbot要求：需要本机使用了白名单Ip（具体深圳白名单IP待咨询，如长沙的白名单IP为公司WiFi的IP，连接WiFi即可搜到），否则搜不到chatbot。

手机设置：具体可看中讯的文档（手机不同有可能会有界面不同，需要找一找设置页面）

手机app出现问题：

解决方法：

清app缓存，重启app

重启手机

卸载重装

关于白名单IP、手机设置以及app方面主要是咨询中讯平台方面的技术支持。

## **二、关于chatbot配置部署**

1.修改 application.properties

需要修改的项：

####### chatbot应用配置 ######  
#chatbot平台地址  
serverRoot=maap.5gmsg-lab.cn:30001  
apiVersion=v1  
chatbotid=2020092901  
##random为文档上的token  
random=mengwang2  
​  
#####第三方介入凭证  
appid=0861812064584ad9b78d633fef25f3a5  
appsecret=fac82caa0efe4570b6d800178df5ba59  
sip=sip:2020092901@botplatform.rcs.chinaunicom.cn  
​

2.修改HttpController

需要修改的地方：

//改成自己的回调地址  
@RequestMapping("/cu/zx/sz/pa")  
​  
//chatbotId修改为自己的chatbotId  
@PostMapping("/messageNotification/sip:2020092901@botplatform.rcs.chinaunicom.cn/messages")

3.nginx配置

nginx.conf:

​  
worker\_processes  1;  
​  
events {  
   worker\_connections  1024;  
}  
​  
​  
http {  
   include       mime.types;  
   default\_type application/octet-stream;  
​  
   sendfile       on;  
   keepalive\_timeout  65;  
​  
    #gzip on;  
​  
   server {  
      
   ## 监听回调地址的ip + 端口  
       listen       37597;     
       server\_name  218.77.105.249;  
​  
        #charset koi8-r;  
​  
        #access\_log logs/host.access.log main;  
          
 ## 平安科技  
       location /cu/zx/sz/pa {     
              #proxy\_pass http://127.0.0.1:37701;  
              ## 代理到服务所在地址  
             proxy\_pass http://192.168.1.172:37701;  
       }  
       ## 益丰大药房  
       location /cu/zx/sz/yf {     
             proxy\_pass http://127.0.0.1:37703;  
              #proxy\_pass http://192.168.1.172:37703;  
       }  
​  
       location / {  
           root   html;  
           index index.html index.htm;  
       }  
​  
        #error\_page 404             /404.html;  
​  
        # redirect server error pages to the static page /50x.html  
        #  
       error\_page   500 502 503 504 /50x.html;  
       location = /50x.html {  
           root   html;  
       }  
​  
   }  
​  
   server {  
       listen       37598;  
       server\_name  218.77.105.249;  
​  
       location /notifications/wltx {     
             proxy\_pass http://127.0.0.1:37702;  
              #proxy\_pass http://192.168.1.172:37702;  
       }  
​  
       location / {  
           root   html;  
           index index.html index.htm;  
       }  
​  
​  
       error\_page   500 502 503 504 /50x.html;  
       location = /50x.html {  
           root   html;  
       }  
   }  
​  
​  
}  
​

## **三、功能：**

已实现：

1）文件上传

2）消息交互，可发送文本、单卡片、多卡片、带按钮以及浮动菜单。

**终端进行功能使用：**

搜索chatbot，进入chatbot，然后发送相应关键字。

如果手机发送成功，没有反应，没有接收到下行。

原因以及解决办法：

nginx没配置好。检查nginx是否配对，查看日志，请求是否到达nginx。

nginx日志状态码如果是200，检查代码的HttpController路径是否配对。

## **四、代码详解 (结合接口文档)**

### **文件上传**

//开启上传图片线程  
scheduledThreadPool.schedule(new FileUploadThread(), 0, TimeUnit.HOURS);  
​  
@Slf4j  
public class FileUploadThread implements Runnable {  
​  
​  
    @SneakyThrows  
    @Override  
    public void run() {  
        ConcurrentMap<String, ReplyContent> map = MakeCardUtil.getReply();  
        if (StringUtils.isBlank(GetToken.token)) {  
            log.info("token为空，上传图片延时5秒");  
            sleep(1000);  
       }  
        Confsing confsing = Conf.getInstance().getConfsing();  
        StringBuilder builder = new StringBuilder("http://");  
        builder.append(confsing.getServerRoot()).append("/bot/")  
               .append(confsing.getApiVersion())  
               .append("/" + confsing.getSip())  
               .append("/medias/upload ");  
        String requestUrl = builder.toString();  
        // Confsing confsing = Conf.getInstance().getConfsing();  
       String path = getRootPath()+confsing.getPicUrl();  
        System.out.println(path);  
        File root = new File(path);  
        File[] files = root.listFiles();  
        try {  
            //循环发送文件  
            for (File f : files) {  
                if (checkImage(f)) {  
                     File thumPic = new File(path + "/sl/" +"thum"+f.getName().replace("png","jpg"));  
                    //发送  
                    String s = SendMsgUtil.mediaFileRequest(requestUrl, null, thumPic, f);  
                    FileResponse fileResponse = parsingUrl(s);  
                    String reChar = "{{" + fileResponse.getFileName() + "}}";  
                    String oldCharThum = "{{sl" + fileResponse.getFileName() + "}}";  
​  
                    String fileSize = "{{mediaFilesize}}";  
                    String thumFileSize = "{{thumbnailFilesize}}";  
​  
                    //遍历模板map，替换图片url  
                    for (Map.Entry<String, ReplyContent> entry : map.entrySet()) {  
                        if (entry.getValue().getContentText() == null) {  
                            continue;  
                       }  
                        String cards = entry.getValue().getContentText().toString();  
                        if (cards.contains(reChar)) {  
                            //替换  
                            cards = cards.replace(reChar, fileResponse.getFileUrl()).replace(oldCharThum, fileResponse.getThumbnailUrl())  
                                   .replace(fileSize,fileResponse.getFileSize()).replace(thumFileSize,fileResponse.getThumFileSize());  
                            JsonObject jsonObject = new JsonParser().parse(cards).getAsJsonObject();  
                            entry.getValue().setContentText(jsonObject);  
                            log.info("替换完成：" + reChar);  
                            break;  
                       }  
                   }  
​  
               }  
           }  
​  
            System.out.println(map);  
       } catch (Exception e) {  
            log.error("缩略图文件未找到", e);  
       }  
​  
   }  
​  
    public String getRootPath(){  
        String jar\_parent = null;  
        try {  
            //拿到项目根路径  
            jar\_parent = new File(ResourceUtils.getURL("classpath:").getPath()).getParentFile().getParentFile().getParent();  
            jar\_parent = java.net.URLDecoder.decode(jar\_parent, "utf-8")+"/";  
            if (jar\_parent.startsWith("file:/")) {  
                jar\_parent = jar\_parent.substring(5);  
           }  
​  
       } catch (Exception e) {  
            log.error("路径错误", e);  
       }  
       // String path = jar\_parent + "/" + confsing.getPicUrl();  
​  
        return jar\_parent;  
   }  
​  
    /\*\*  
     \* 检查是否为图片文件  
     \*  
     \* @param file  
     \* @return  
     \*/  
    public static boolean checkImage(File file) {  
        try {  
            //通过图片io流判断  
            Image image = ImageIO.read(file);  
            return image != null;  
       } catch (IOException e) {  
            log.error("该文件不是图片类型:" + file.getName());  
            return false;  
​  
       }

### **消息交互**

整个过程主要是先接收手机终端发送的上行消息，然后拿到关键字进行消息匹配，发送对应下行消息。

接收消息代码：

HttpController:

@Slf4j  
@RestController  
@RequestMapping("/cu/zx/sz/pa")  
public class HttpController {  
​  
    @Autowired  
    private HttpService httpService;  
​  
    /\*\*  
     \* 接收消息  
     \*  
     \* @param receMessage  
     \* @return  
     \*/  
    @PostMapping("/messageNotification/sip:2020092901@botplatform.rcs.chinaunicom.cn/messages")  
   // @RequestMapping(path = "/messageNotification/\*", method = RequestMethod.POST)  
    @ResponseBody  
    public String receiveMessage(@RequestBody ReceMessage receMessage) {  
        httpService.receiveMessageService(receMessage);  
        return null;  
   }

HttpService:

public interface HttpService {  
    /\*\*\*  
     \* 处理接收消息  
     \* @param receMessage  
     \*/  
    void receiveMessageService(ReceMessage receMessage);

HttpServiceImpl:

@Service  
@Slf4j  
public class HttpServiceImpl implements HttpService {  
​  
    @Autowired  
    public Confsing confsing;  
    /\*\*  
     \* 下行详单  
     \*/  
    private static Long sendFileName;  
    /\*\*  
     \* 上行详单  
     \*/  
    private static Long recvFileName;  
    /\*\*  
     \* 状态报告详单  
     \*/  
    private static Long statusFileName;  
​  
    /\*\*\*  
     \* 处理接收消息  
     \* @param receMessage  
     \*/  
    @Override  
    public void receiveMessageService(ReceMessage receMessage) {  
        //解析上行报文获取关键字  
        try {  
            Object[] b = receMessage.getMessageList();  
            ObjectMapper objectMapper = new ObjectMapper();  
            MessageList messageList = objectMapper.convertValue(b[0], MessageList.class);  
            Object obj = messageList.getContentText();  
            String content;  
​  
            /\*\*匹配下行处理start\*\*/  
            if ("text/plain".equals(messageList.getContentType())) {  
                content = (String) obj;  
           } else if ("application/vnd.gsma.botsuggestion.response.v1.0+json".equals(messageList.getContentType())) {  
                String errStr = JSONObject.toJSONString(obj);  
                JSONObject parseObject = JSONObject.parseObject(errStr);  
                JSONObject response = (JSONObject) parseObject.get("response");  
                JSONObject reply = (JSONObject) response.get("reply");  
                if(reply == null){  
                    return;  
               }  
                content = reply.getString("displayText");  
           } else {  
                content = "default";  
           }  
            log.info("上行消息关键字为:{},手机号:{}",content,receMessage.getSenderAddress());  
              
            //通过关键字进行消息匹配，然后组装发送消息对象  
            MessageBody messageBody = MakeCardUtil.getSendMessage(receMessage, content);  
//           Object obj1 = JSONArray.toJSON(messageBody);  
//           String sendMessage = obj1.toString();  
          // String jsonStr = JSONObject.toJSONString(object);  
            GsonBuilder gsonBuilder = new GsonBuilder();  
            gsonBuilder.setPrettyPrinting();  
            Gson gson = gsonBuilder.create();  
            String sendMessage = gson.toJson(messageBody);  
            //String sendMessage = JSONObject.toJSONString(messageBody);  
           //String sendMessage = JSON.toJSONString(messageBody);  
            /\*\*匹配下行处理end\*\*/  
​  
            //填写上行详单  
         //   writeRecv(receMessage);  
​  
            StringBuilder builder = new StringBuilder("http://");  
            builder.append(confsing.getServerRoot()).append("/bot/")  
                   .append(confsing.getApiVersion())  
                   .append("/" + confsing.getSip())  
                   .append("/messages");  
            String requestUrl = builder.toString();  
​  
            String request = SendMsgUtil.request(requestUrl, sendMessage);  
            System.out.println(request);  
            //writeSend(messageBody);  
       } catch (Exception e) {  
            log.error("处理接收消息异常", e);  
       }  
   }

通过上行中的关键字进行消息匹配，然后组装发送消息对象：

/\*\*  
     \* 上行关键字匹配处理  
     \* @param receMessage  
     \* @param bodyText  
     \* @return  
     \*/  
    public static ReplyContent getReplyContent(ReceMessage receMessage, String bodyText){  
        ConcurrentHashMap<String, Integer> userMap = GlobalUserDataConfig.userdata;  
        //创建用户个人的上下文map  
        if(!userMap.containsKey(receMessage.getSenderAddress())){  
            userMap.put(receMessage.getSenderAddress() , 0);  
       }  
​  
        if(userMap.get(receMessage.getSenderAddress()).equals(0)){  
            if("投资/贷款/理财/投资/贷款/保险/炒股".contains(bodyText)){  
                bodyText = "投资";  
           }else if("热门活动".contains(bodyText)){  
                bodyText = "热门活动";  
           }else if("账户总览/余额".contains(bodyText)){  
                bodyText = "发送数字";  
                //进入上下文关系  
                userMap.put(receMessage.getSenderAddress() , 1);  
           }else if("信用卡/信用卡账单".contains(bodyText)){  
                bodyText = "信用卡";  
           }else if("更多业务办理".contains(bodyText)){  
                bodyText = "更多业务";  
           }else if("精选大礼".contains(bodyText)){  
                bodyText = "精选大礼";  
           }else if("工资理财".contains(bodyText)){  
                bodyText = "工资理财";  
           }else if("我要借款".contains(bodyText)){  
                bodyText = "我要借款";  
           }else{  
                bodyText = "default";  
           }  
       }else {  
            //上下文关系  
            if(bodyText!=null && bodyText.matches("^[\\d]{4}$")){  
                bodyText = "已登录";  
                userMap.put(receMessage.getSenderAddress() , 0);  
           }else if("取消".contains(bodyText)){  
                bodyText = "取消";  
                userMap.put(receMessage.getSenderAddress() , 0);  
           }else if("退出".contains(bodyText)){  
                bodyText = "退出";  
                userMap.put(receMessage.getSenderAddress() , 0);  
           }else{  
                bodyText = "数字错误";  
           }  
       }  
​  
        ReplyContent replyContent = reply.get(bodyText);  
​  
        if (replyContent == null) {  
            replyContent = reply.get("default");  
       }  
        if(!("取消".equals(bodyText)) && !("退出".equals(bodyText))){  
            replyContent.setContentText(JSONObject.parse(replyContent.getContentText().toString()));  
       }  
​  
        return replyContent;  
   }

获取发送对象：

/\*\*  
     \* 获取发送对象  
     \*  
     \* @param receMessage  
     \* @param bodyText  
     \* @return  
     \*/  
    public static MessageBody getSendMessage(ReceMessage receMessage, String bodyText) {  
​  
        try {  
            ReplyContent replyContent = getReplyContent(receMessage, bodyText);  
​  
            ServiceCapability serviceCapability = new ServiceCapability();  
            serviceCapability.setVersion("+g.gsma.rcs.botversion=&quot;#=1&quot;");  
            serviceCapability.setCapabilityId("ChatbotSA");  
​  
            MessageBody messageBody = new MessageBody();  
            messageBody.setMessageId(UUID.randomUUID().toString());  
            messageBody.setDestinationAddress(new String[]{receMessage.getSenderAddress()});  
            messageBody.setSenderAddress(receMessage.getDestinationAddress());  
            messageBody.setConversationId(receMessage.getConversationId());  
            messageBody.setContributionId(receMessage.getContributionId());  
            Object[] c = {JSON.toJSON(serviceCapability)};  
            messageBody.setServiceCapability(c);  
           // messageBody.setSmsSupported(true);  
            messageBody.setImFormat("IM");  
            messageBody.setInReplyTo(receMessage.getContributionId());  
            String[] report = {"sent", "delivered"};  
            messageBody.setReportRequest(report);  
           // String sendMessage = JSON.toJSONString(floatingMenu.getSuggestions());  
          // messageBody.setSmsContent(JSON.toJSONString(replyContent.getContentText()));  
          // Object parse = JSONObject.parse((String) floatingMenu.getSuggestions());  
           // replyContent.setSuggestions(floatingMenu.getSuggestions());  
            ReplyContent suggestion;  
            ReplyContent message = new ReplyContent(replyContent.getContentText(),replyContent.getContentType(),null);  
            ReplyContent[] messageList;  
            System.out.println( replyContent.getSuggestions());  
​  
            if(replyContent.getSuggestions() != null){  
                suggestion = new ReplyContent();  
                JSONObject jsonObject = new JSONObject();  
                jsonObject.put("suggestions",replyContent.getSuggestions());  
                suggestion.setContentText(jsonObject);  
                suggestion.setContentType("application/vnd.gsma.botsuggestion.v1.0+json");  
                messageList = new ReplyContent[]{message, suggestion};  
           }else {  
             messageList = new ReplyContent[]{message};  
           }  
            messageBody.setMessageList(messageList);  
​  
            //System.out.println(JSON.toJSONString(messageBody));  
            return messageBody;  
​  
       } catch (Exception e) {  
            log.error("获取发送内容异常", e);  
       }  
        return null;  
   }

发送消息代码：

/\*\*  
     \* 发送请求   httpclient连接池,模板初始化  
     \*  
     \* @param requestUrl  
     \* @param  
     \*/  
    public static String request(String requestUrl, String messageBody) {  
        try {  
​  
            HttpHeaders headers = new HttpHeaders();  
            headers.setContentType(MediaType.APPLICATION\_JSON);  
            String gmt = CommonUtil.getGMT();  
            headers.add("Date", gmt);  
            HttpEntity httpEntity;  
            if (StringUtils.isNoneBlank(messageBody)) {  
​  
                if (StringUtils.isBlank(GetToken.token)) {  
                    log.info("token为空，稍后再试");  
                    //   WipingTheButt.fecesOffer(new Fece(messageBody, requestUrl));  
                    return null;  
               }  
​  
                headers.add("Authorization", "accessToken "+ GetToken.token);  
                httpEntity = new HttpEntity(JSONObject.parseObject(messageBody), headers);  
​  
           } else {  
​  
                Confsing confsing = Conf.getInstance().getConfsing();  
                Map<String, String> requestBody = new HashMap();  
                requestBody.put("appId", confsing.getAppid());  
                requestBody.put("appKey", confsing.getAppsecret());  
                httpEntity = new HttpEntity(requestBody, headers);  
           }  
            log.info("请求地址:" + requestUrl);  
            log.info("headers:" + headers.toString());  
            log.info("向5G消息发送接口请求报文为：{}", JSONObject.parse(messageBody));  
            ResponseEntity<Object> response = sendMsg.restTemplate.postForEntity(requestUrl, httpEntity, Object.class);  
            if (!response.getStatusCode().equals(HttpStatus.OK) || response.getBody() == null) {  
                log.error("5G消息下行异常, URL地址:{}, 状态:{}", requestUrl, response.getStatusCode());  
           }  
​  
            log.info("消息下行成功");  
            return JSON.toJSONString(response.getBody());  
       } catch (ResourceAccessException e) {  
            log.error("SendMsgUtil.request:发送消息产生网络异常");  
            WipingTheButt.fecesOffer(new Fece(messageBody, requestUrl));  
       } catch (Exception e) {  
            log.error("发送消息异常", e);  
       }  
        return null;  
   }

获取回复内容代码：

//加载卡片和回复  
        try {  
            MakeCardUtil.setReply(getCardsAndReply());  
            log.info("加载卡片和回复成功 Card loading and reply successful");  
              
              
/\*\*  
     \* 获取卡片和回复  
     \*  
     \* @throws IOException  
     \* @throws RuntimeException  
     \*/  
    public static Map getCardsAndReply() throws Exception {  
​  
        String s = FileUtils.readFileToString(ResourceUtils.getFile("classpath:cardsAndReply.txt"), "utf-8");  
        if (StringUtils.isEmpty(s)) {  
            log.error("cardsAndReply为空");  
            throw new RuntimeException();  
       }  
​  
        Map<String, ReplyContent> reply = new HashMap<>();  
​  
        JSONArray jsonArray = JSONArray.parseArray(s);  
​  
        // JsonArray jsonArray = new JsonParser().parse(s).getAsJsonArray();  
        ReplyContent replyContent;  
        String recvText;  
        String type;  
        Object content = null;  
        Object suggestions;  
        for (Object j : jsonArray) {  
            JSONObject object = (JSONObject) j;  
​  
            recvText = object.getString("recvText");  
            type = object.getString("contentType");  
​  
            if ("text/plain".equals(type)) {  
                content = object.getString("text");  
           }  
​  
            if("application/vnd.gsma.botmessage.v1.0+json".equals(type)){  
                content = object.getString("cards");  
           }  
            suggestions = object.get("suggestions");  
            replyContent = new ReplyContent(content, type, suggestions);  
            reply.put(recvText, replyContent);  
       }  
        System.out.println(reply);  
        return reply;  
​  
   }

## **五、需求开发**

先根据需求修改回复模板文件cardsAndReply.txt

再修改上行关键字匹配处理代码：

/\*\*  
     \* 上行关键字匹配处理  
     \* @param receMessage  
     \* @param bodyText  
     \* @return  
     \*/  
    public static ReplyContent getReplyContent(ReceMessage receMessage, String bodyText){  
        ConcurrentHashMap<String, Integer> userMap = GlobalUserDataConfig.userdata;  
        //创建用户个人的上下文map  
        if(!userMap.containsKey(receMessage.getSenderAddress())){  
            userMap.put(receMessage.getSenderAddress() , 0);  
       }  
​  
        if(userMap.get(receMessage.getSenderAddress()).equals(0)){  
            if("投资/贷款/理财/投资/贷款/保险/炒股".contains(bodyText)){  
                bodyText = "投资";  
           }else if("热门活动".contains(bodyText)){  
                bodyText = "热门活动";  
           }else if("账户总览/余额".contains(bodyText)){  
                bodyText = "发送数字";  
                //进入上下文关系  
                userMap.put(receMessage.getSenderAddress() , 1);  
           }else if("信用卡/信用卡账单".contains(bodyText)){  
                bodyText = "信用卡";  
           }else if("更多业务办理".contains(bodyText)){  
                bodyText = "更多业务";  
           }else if("精选大礼".contains(bodyText)){  
                bodyText = "精选大礼";  
           }else if("工资理财".contains(bodyText)){  
                bodyText = "工资理财";  
           }else if("我要借款".contains(bodyText)){  
                bodyText = "我要借款";  
           }else{  
                bodyText = "default";  
           }  
       }else {  
            //上下文关系  
            if(bodyText!=null && bodyText.matches("^[\\d]{4}$")){  
                bodyText = "已登录";  
                userMap.put(receMessage.getSenderAddress() , 0);  
           }else if("取消".contains(bodyText)){  
                bodyText = "取消";  
                userMap.put(receMessage.getSenderAddress() , 0);  
           }else if("退出".contains(bodyText)){  
                bodyText = "退出";  
                userMap.put(receMessage.getSenderAddress() , 0);  
           }else{  
                bodyText = "数字错误";  
           }  
       }  
​  
        ReplyContent replyContent = reply.get(bodyText);  
​  
        if (replyContent == null) {  
            replyContent = reply.get("default");  
       }  
        if(!("取消".equals(bodyText)) && !("退出".equals(bodyText))){  
            replyContent.setContentText(JSONObject.parse(replyContent.getContentText().toString()));  
       }  
​  
        return replyContent;  
   }

## **六、有问题可以联系我**

QQ：1074623424

电话：13786380232

长沙项目三部--周鹏