1. 腾讯IM发送emoji表情部分

相关文件：



1、在ChatActivity中的sendText()方法内新建TextMessage对象，通过ChatPresenter对象的sendMessage()方法来发送消息。TextMessage()对象的构造方法内填入输入框的内容，内容通过ChatInput控件的getText()方法来获取。

2、ChatInput控件是底部消息栏的自定义控件，getText()方法中通过调用自己实现的ReplaceFun3()方法来重新更改EditText框中的内容。

3、ReplaceFun3（String s）方法 （一点一点的拼接EditText）

需要将[可爱]这样的字符串更改为emoji表情，主要实现是先通过正则表达式删选出是[可爱]这种类型的字符串，在循环中，通过自己实现的indexPosition（matcher.group）方法来检索筛选出来的字符串在EmoticonUtil中的数组中的位置，若>-1，则存在该表情，将该表情拼接到EditText中。

1. 表情是由SpannableString来实现的，先通过AssetManager从数据库中获取到表情，通过一系列操作形成ImageSpan对象，然后通过SpannableString对象的setSpan（）方法来传入该ImageSpan对象。

！！！此处的坑是，新建的SpannableString对象传入的String是该emoji的下标通过String.valueof（index）方法来得到的，然后setSpan（）的起始长度是该string的长度。

*//\*全部成功，拼凑内容***private void** ReplaceFun3(String comstr){  
 String EMOJI = **"\\[[\\u4e00-\\u9fa5a-zA-Z]+]"**;  
 SpannableString spannableString = **new** SpannableString(comstr);*//\*整体内容* Pattern pattern = Pattern.*compile*(EMOJI);  
 Matcher matcher = pattern.matcher(spannableString);  
 **boolean** result = matcher.find();  
 **int** start=0; *//供下次检索到emoji来加入两个emoji之间的内容  
  
 //循环直到匹配不到* **if**(result){  
 **editText**.setText(**""**);  
 **editText**.append(spannableString.subSequence(0,matcher.start()));*//第一次匹配到，加入该emoji前的字符串* start=matcher.start();  
 }  
 **while** (result) { *//循环判断是否含有[可爱]类型的字符串* **int** index = indexPosition(matcher.group()); *//\*表情在EmoticonUtil数组中的位置* **int** end = matcher.end(); *//供最后一次检索到emoji，加入该emoji后的字符串使用* **if**(index > -1){ *//含有此表情* **try** {  
 AssetManager am = getContext().getAssets(); *//读取数据库部分* InputStream is = am.open(String.*format*(**"emoticon/%d.png"**, index));  
 Bitmap bitmap = BitmapFactory.*decodeStream*(is);  
 Matrix matrix = **new** Matrix();  
 **int** width = bitmap.getWidth();  
 **int** height = bitmap.getHeight();  
 **float** s = ScreenUtil.*instance*(**context**).getDensity(**context**);  
 **if** (s >= 3)  
 matrix.postScale(2.5f, 2.5f);  
 **else** matrix.postScale(1.3f, 1.3f);  
 **final** Bitmap resizedBitmap = Bitmap.*createBitmap*(bitmap, 0, 0,  
 width, height, matrix, **true**);  
 ImageSpan span = **new** ImageSpan(getContext(), resizedBitmap, ImageSpan.***ALIGN\_BASELINE***);  
  
 String content = String.*valueOf*(index);  
 SpannableString emojiString = **new** SpannableString((String.*valueOf*(index)));*//\*只是emoji表情部分  
 //matcher.start() matcher.end()是匹配到的字符串在原始字符串中的起始位置，进行替换* emojiString.setSpan(span, 0, content.length(), Spanned.***SPAN\_INCLUSIVE\_EXCLUSIVE***); *//\*被替换的内容,长度为数字转化的字符串长度* **editText**.append(spannableString.subSequence(start,matcher.start())); *//加入emoji表情间的普通字符串* **editText**.append(emojiString); *//加入emoji字符串* start=matcher.end();  
 result = matcher.find(); *//为false后后面就没有emoji了* **if**(result==**false**){  
 **editText**.append(spannableString.subSequence(end,spannableString.length())); *//\*最后一个表情，加入其后的普通字符串* }  
 is.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 **else** { *//不含有此表情，也需要加上此字符串* **editText**.append(spannableString.subSequence(start,matcher.start()));  
 result = matcher.find();  
 }  
 }  
}

*//\*检索表情在数组中的下标 已成功，有用***public int** indexPosition(String str){  
 **for**(**int** i = 0; i< EmoticonUtil.*emoticonData*.**length**; i++){  
 **if**(EmoticonUtil.*emoticonData*[i].equals(str)){  
 **return** i;  
 }  
 }  
 **return** -1;  
}

1. 发送给朋友中emoji文本转化部分

使用的类：



1. 复制、删除、转发的弹出框在ChatAdapter.java中的messageDisPop()方法中展现，该方法被disposeInfo()调用，disposeInfo()方法在长按时被调用。
2. messageDisPop()方法中，当点击发送给朋友时会掉转至FriendTransmitActivity活动中，当选择头像时，在OnItemClick()方法里，通过自己抽取出的MessageUtil类的convertEmoji()方法来将信息进行转化传出Editable对象，将该对象赋给TextMessage()构造方法即可。
3. 输入框位置更改问题

在ChatInput.java中，在updateView()方法中，点击MORE按钮，根据现在morePanel是否显示来控制其再次点击是隐藏还是显示。在chat\_input.xml中，将morePanel布局初始设置为gone

1. 个人聊天记录的清空

**旧的麻烦的方法**

1. 首先先从ChatDetailsUI中传来identify和TIMConversation，用这两个凭证来搜索聊天记录。通过ChatPresenter对象的getMessage()方法来获取，搜索后会调用showMessage()方法，在这个方法里给自定义的List<Message>集合赋值，在点击清空按钮时，通过将list集合中每个Message对象调用remove()方法来实现将聊天记录删除，删除完成后向上一级活动返回isclean标识，

2、在ChatActivity活动的onActivityResult（）方法中，根据返回来的boolean判断聊天记录是否清空了，清空了则将该活动中的messageList集合清空并更新适配器，即可达到界面上清空聊天记录的实现。

**新的实现方法**

聊天记录的清空实现已经更改（更简便）：

直接在onActivityResult（）方法内将ChatActivity中的messageList集合中的message循环remove（）清除，然后messageList通过clean（）清除，更新listView即可。

1. 群聊聊天记录的清空

直接在onActivityResult（）方法内将ChatActivity中的messageList集合中的message循环remove（）清除，然后messageList通过clean（）清除，更新listView即可。

ChatView聊天界面接口，聊天界面继承他，通过ChatPresenter对象的 presenter.getMessage(messageList.size() > 0 ? messageList.get(0).getMessage() : null);方法来请求数据，请求完会回调showMessage()方法，在此方法内将获取到的聊天记录集合赋给本地集合，然后将集合内的Message元素通过remove()方法来清除该条记录的缓存，这就相当于清掉了这一条聊天记录。