# 第6章消息推送

在此项目中我们用到了极光推送，具体的用法请参照官网。

### *快速集成*

1. 快速集成网址：http://docs.jpush.io/guideline/android_guide/
2. 参照：Android 集成指南 - 极光文档.htm
3. 注意点：

   默认下载的只能在arm的模拟器下运行。appkey和应用程序的包名，必须和你申请的程序保证一致。

4. 如何获取通知内的参数

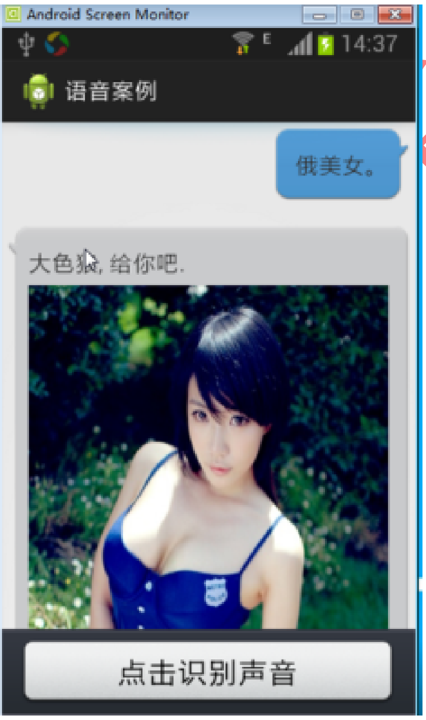
   服务器在推送过通知后，会在通知内带一些数据，通过这些数据我们可以做相应的操作。那么如何获取？通过广播接受者

```
<!-- User defined.    用户自定义的广播接收器-->
<receiver
      android:name="com.itheima41.msgpush.MyReceiver"
      android:enabled="true">
      <intent-filter>
<!--  <action android:name="cn.jpush.android.intent.REGISTRATION"
/> Required  用户注册SDK的intent -->
<!--  <action
android:name="cn.jpush.android.intent.UNREGISTRATION" />   -->
<!--  <action
android:name="cn.jpush.android.intent.MESSAGE_RECEIVED" />
Required  用户接收SDK消息的intent -->
<!--  <action
android:name="cn.jpush.android.intent.NOTIFICATION_RECEIVED" />
Required  用户接收SDK通知栏信息的intent -->
      <action
android:name="cn.jpush.android.intent.NOTIFICATION_OPENED" /> <!-
-Required  用户打开自定义通知栏的intent-->
<!--  <action android:name="cn.jpush.android.intent.CONNECTION"
/>接收网络变化 连接/断开 since 1.6.3 -->
            <category android:name="com.itheima41.msgpush" />
      </intent-filter>
</receiver>

public class MyReceiver extends BroadcastReceiver {
      @Override
      public void onReceive(Context context, Intent intent) {
            // 取出服务器推送过来的值.
            Bundle bundle = intent.getExtras();
            String url =
bundle.getString(JPushInterface.EXTRA_EXTRA);
            System.out.println("内容url: " + url);
      }
}
```

# 第7章语音案例

在做语音识别方面，我们用了科大讯飞的语音识别引擎。如图：“点击识别声音”，说：“美女”，程序识别后就返回一个美女图片。



俄美女。

大色狼, 给你吧.

## 语音布局实现

这个布局类似于qq聊天，自己说话在右边，对方回复在左侧。中间的说话与回答功能，可以用listview。item可以是左右两个TextView，然后判断是问题还是回答，来区分左右显示。

子布局代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="5dip" >

    <LinearLayout
        android:id="@+id/ll_answer"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/answer_bubble"
        android:orientation="vertical"
        android:padding="10dip"
        android:visibility="gone" >

        <TextView
            android:id="@+id/tv_answer_text"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_margin="5dip"
            android:text="没听明白."
            android:textColor="#000000"
            android:textSize="18sp" />

        <ImageView
            android:id="@+id/iv_answer_image"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:maxHeight="200dip"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_launcher" />
    </LinearLayout>
    <TextView
        android:id="@+id/tv_asker_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:background="@drawable/asker_bubble"
        android:padding="10dip"
        android:text="你好"
        android:textColor="#000000"
        android:textSize="18sp"
        android:visibility="gone" />
</LinearLayout>
```

## 逻辑实现

- JavaBean

```
public class ConversationInfoBean {

    public static final int ASKER = 0; // 提问者
    public static final int ANSWER = 1; // 回答者

    public String text; // 文本, 一定共有.
    public int imageID = -1; // 图片的id, 可能有. 当等于-1时,
是没有图片展示的
    public int who;

    public ConversationInfoBean() {
        super();
    }

    public ConversationInfoBean(String text, int imageID, int
who) {
        super();
        this.text = text;
        this.imageID = imageID;
        this.who = who;
    }

    public String getText() {
        return text;
    }

    public void setText(String text) {
        this.text = text;
    }

    public int getImageID() {
        return imageID;
    }

    public void setImageID(int imageID) {
        this.imageID = imageID;
    }

    public int getWho() {
        return who;
        }

        public void setWho(int who) {
                this.who = who;
        }
}
```

- 语音识别

```
class MyRecognizerDialogListener implements
RecognizerDialogListener {

        private String builder = "";

        /**
         * isFinish 当前语音是否识别并解析完成, false 没有完成,
true这次识别完成了
         */
        @Override
        public void onResult(RecognizerResult result, boolean
isFinish) {
                System.out.println("onResult: " + isFinish);
                String text = result.getResultString();
                text = processJSON(text);
                System.out.println("听到: " + text);
                builder = builder + text;//拼接识别出的内容
                if(!isFinish) { // 当前未识别完毕
                        return;
                }
                // text 当前这个text是把声音转换成的文字, 是提问者的数据
                ConversationInfoBean bean = new
ConversationInfoBean(builder, -1, ConversationInfoBean.ASKER);
                builder = "";//重新初始化
                conversationList.add(bean);//添加到集合中
                mAdapter.notifyDataSetChanged();//更新listView
                String answerText = "没听明白.";
                int answerImageID = -1;
                Random random = new Random();
                // 依赖问的问题, 进行选择回答的内容
                if(bean.text.contains("美女")) {
                        int index =
random.nextInt(ResourceUtils.mmArray.length);
                        answerText = ResourceUtils.mmTextArray[index];
                        answerImageID = ResourceUtils.mmArray[index];
                } else if(bean.text.contains("老师")) {
```

```
            int index = 
random.nextInt(ResourceUtils.teacherArray.length);
            answerText = 
ResourceUtils.teacherTextArray[index];
            answerImageID = 
ResourceUtils.teacherArray[index];
        } else if(bean.text.contains("天王盖地虎")) {
            int index = 
random.nextInt(ResourceUtils.markArray.length);
            answerText = ResourceUtils.markTextArray[index];
            answerImageID = ResourceUtils.markArray[index];
        }
        bean = new ConversationInfoBean(answerText, 
answerImageID, ConversationInfoBean.ANSWER);
        conversationList.add(bean);
        mAdapter.notifyDataSetChanged();
        mListView.setSelection(mListView.getCount());//定位
        // 播放声音
        mSpeechUtils.speakLanguage(answerText);
    }
    @Override
    public void onError(SpeechError arg0) {
    }
}
```

- 解析json数据

```
    private String processJSON(String json) {
        Gson gson = new Gson();
        VoiceBean bean = gson.fromJson(json, VoiceBean.class);
        return bean.ws.get(0).cw.get(0).w;
    }
```

- VoiceBean

```
public class VoiceBean {

    public String bg;
    public String ed;
    public String ls;
    public String sn;
    public List<WS> ws;

    public class WS {

        public String bg;
        public List<CW> cw;
    }
```

```
      public class CW {
            public String sc;
            public String w;
      }
}
```

- listView适配

```
class ConversationAdapter extends BaseAdapter {

      @Override
      public int getCount() {
            return conversationList.size();
      }

      @Override
      public View getView(int position, View convertView,
ViewGroup parent) {
            View view = null;
            if(convertView == null) {
                  view = View.inflate(MainActivity.this,
R.layout.listview_item, null);
            } else {
                  view = convertView;
            }

            ConversationInfoBean bean =
conversationList.get(position);

            View answer = view.findViewById(R.id.ll_answer);
            TextView tvAskerText = (TextView)
view.findViewById(R.id.tv_asker_text);
            if(bean.who == ConversationInfoBean.ASKER) {//如果是提问
                  answer.setVisibility(View.GONE);
                  tvAskerText.setVisibility(View.VISIBLE);
                  tvAskerText.setText(bean.text);
            } else {//如果是回答
                  answer.setVisibility(View.VISIBLE);
                  tvAskerText.setVisibility(View.GONE);
                  TextView tvAnswer = (TextView)
view.findViewById(R.id.tv_answer_text);
                  tvAnswer.setText(bean.text);
                  ImageView ivImage = (ImageView)
view.findViewById(R.id.iv_answer_image);
                  if(bean.imageID == -1) {
```

```
                // 当前这条会话不设置图片，也不能让imageview显示
                ivImage.setVisibility(View.GONE);
            } else {
                ivImage.setVisibility(View.VISIBLE);
                ivImage.setImageResource(bean.imageID);
            }
        }
        return view;
    }

    @Override
    public Object getItem(int position) {
        return null;
    }
    @Override
    public long getItemId(int position) {
        return 0;
    }
}
```