

智能聊天机器人

一、搭建界面布局

1. 搭建主界面布局

编写activity_main.xml，父布局使用LinearLayout。

```
<TextView
    android:id="@+id/tvTitle"
    android:gravity="center"
    android:textSize="20sp"
    android:textColor="@color/B_group_1"
    android:background="@color/B_group_3"
    android:text="机器人"
    android:layout_width="match_parent"
    android:layout_height="45sp"></TextView>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ListView
        android:id="@+id/lvContent"
        android:layout_above="@id/llBottom"
        android:divider="@null"
        android:listSelector="@null"
        android:transcriptMode="alwaysScroll"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"></ListView>
    <LinearLayout
        android:id="@+id/llBottom"
        android:padding="10dp"
        android:background="@color/B_group_3"
        android:layout_alignParentBottom="true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <EditText
            android:id="@+id/etContent"
            android:layout_weight="4"
            android:layout_marginRight="5dp"
            android:layout_width="0dp"
            android:layout_height="wrap_content"></EditText>
        <Button
            android:id="@+id/btSend"
            android:text="发送"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"></Button>
    </LinearLayout>
</RelativeLayout>
```

创建背景选择器btn_send_selector，当点击发送按钮时，图片背景会有所区别

```
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@color/B_group_4" android:state_pressed="true">
</item>
    <item android:drawable="@color/B_group_2"></item>
</selector>
```

2. 搭建聊天条目布局

创建list_item_left.xml，父布局使用RelativeLayout，作为用户聊天信息布局

```
<ImageView
    android:id="@+id/iv_head"
    android:layout_width="65dp"
    android:layout_height="65dp"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="5dp"
    android:background="@drawable/my_head"
    android:focusable="false" />
<TextView
    android:id="@+id/tv_chat_content"
    style="@style/chat_content_style"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/B_group_4"
    android:layout_marginRight="5dp"
    android:layout_toLeftOf="@id/iv_head"
    android:background="@drawable/chat_right_bg_normal" />
```

创建list_item_right.xml，父布局使用RelativeLayout，作为接收的机器人信息布局

```
<ImageView
    android:id="@+id/iv_head"
    android:layout_width="65dp"
    android:layout_height="65dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="5dp"
    android:background="@drawable/robot_head"
    android:focusable="false" />
<TextView
    android:id="@+id/tv_chat_content"
    style="@style/chat_content_style"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/B_group_3"
    android:layout_marginLeft="5dp"
    android:layout_toRightOf="@id/iv_head"
    android:background="@drawable/chat_left_bg_normal" />
```

在styles.xml中编写以下代码，作为聊天文本信息属性

lineSpacingExtra:行间距

```
<style name="chat_content_style">
    <item name="android:minHeight">50dp</item>
    <item name="android:gravity">center_vertical</item>
    <item name="android:layout_marginTop">12dp</item>
    <item name="android:textColor">@color/B_group_1</item>
    <item name="android:textSize">15sp</item>
    <item name="android:linespacingExtra">2dp</item>
    <item name="background">@color/B_group_4</item>
</style>
```

二、实现智能机器人通信

1. 封装聊天实体类

创建ChatBean类，定义消息的状态、消息的内容等属性，并创建get、set和toString方法

```
public static final int SEND=1;    //发送消息
public static final int RECEIVE=2;  //接收消息
private int state;    //消息的状态（发送还是接收）
private String message;    //消息的内容
public ChatBean(int state, String message) {
    this.state = state;
    this.message = message;
}
public int getState() {
    return state;
}
public void setState(int state) {
    this.state = state;
}
public String getMessage() {
    return message;
}
public void setMessage(String message) {
    this.message = message;
}
@Override
public String toString() {
    return "ChatBean{" +
        "state=" + state +
        ", message='" + message + '\'' +
        '}';
}
```

2. 设置机器人的欢迎信息

在res/strings.xml中编写以下信息，用于运行程序时显示。

```

<string-array name="welcome">
    <item>亲爱的，想死我了，么么哒，(๑_๑)</item>
    <item>"更多资源，请访问：http://myapptg.com</item>
    <item>主人，人家好无聊呀，来陪我要吧</item>
    <item>主人，您又来找人要了呀，不能要太久哦，您还要好好复习呢</item>
    <item>主人，自从使用了高考助手，成绩是不是提高了呢？</item>
    <item>小主人，你是不是学习累了，您可以输入，来个笑话，我可以为您讲笑话哦</item>
    <item>主人，你是不是上班疲惫了？您可以输入，讲故事，我可以为您讲一些有趣的故事哟</item>
    <item>今天，我好累，不想和你聊天</item>
    <item>小主人您复习的怎么样了</item>
</string-array>

```

3. 编写聊天列表适配器

用于对ListView进行数据适配，从而显示聊天信息。

```

private Context context;
private List<ChatBean> chatBeanList;

public ChatAdapter(Context context, List<ChatBean> chatBeanList) {
    this.context = context;
    this.chatBeanList = chatBeanList;
}

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

@Override
public Object getItem(int position) {
    return chatBeanList.get(position);
}

@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder viewHolder=new ViewHolder();
    ChatBean chatBean = chatBeanList.get(position);
    if (chatBean.getState()==ChatBean.SEND){
        //判断是发送的消息还是接收的消息，如果是发送消息就显示右布局，如果是接收消息就显示左布局
        convertView=View.inflate(context,R.layout.list_item_right,null);
    }else {
        convertView=View.inflate(context,R.layout.list_item_left,null);
    }
    viewHolder.tvContent=convertView.findViewById(R.id.tv_chat_content);
    viewHolder.tvContent.setText(chatBean.getMessage());
    return convertView;
}

class ViewHolder{
    TextView tvContent;
}

```

4. 编写页面交互代码

```
public class MainActivity extends AppCompatActivity {
    private ListView lvContent;
    private EditText etContent;
    private Button btSend;

    private List<ChatBean> chatBeanList;
    private ChatAdapter chatAdapter;

    private String sendMsg;
    private ChatBean chatBean;

    private String welcome[];
    private static final String WEB_SITE =
"http://www.tuling123.com/openapi/api";
    private static final String KEY = "3e7ac6cff6b64a7d939d09a6d6c29642";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        lvContent = (ListView) findViewById(R.id.lvContent);
        etContent = (EditText) findViewById(R.id.etContent);
        btSend = (Button) findViewById(R.id.btSend);
        welcome=getResources().getStringArray(R.array.welcome); //获取欢迎信息
        chatBeanList=new ArrayList<>();
        chatAdapter=new ChatAdapter(this,chatBeanList);
        lvContent.setAdapter(chatAdapter);
        //通过随机数来决定欢迎信息
        int position=(int)(Math.random()*welcome.length-1);
        showData(welcome[position]);
        //发送按钮点击事件
        btSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendMsg=etContent.getText().toString();
                if (TextUtils.isEmpty(sendMsg)){ //如果输入框中无数据,提示输入为空
                    Toast.makeText(this,"输入为空",Toast.LENGTH_SHORT).show();
                    return;
                }
                //获取发送数据之后,清空输入框,
                etContent.setText("");
                //处理发送数据,替换空格和换行
                sendMsg=sendMsg.replaceAll(" ","").replaceAll("\n","").trim();
                chatBean=new ChatBean(ChatBean.RECEIVE,sendMsg);
                chatBeanList.add(chatBean);
                chatAdapter.notifyDataSetChanged();
                getDataFromServer();// 调用方法获取服务器数据
            }
        });
    }

    //适配器更新,显示发送的消息
    private void showData(String msg) {
        ChatBean chatBean=new ChatBean(ChatBean.SEND,msg);
        chatBeanList.add(chatBean);
    }
}
```

```

        Log.i("Aye-showData", chatBean.toString());
        chatAdapter.notifyDataSetChanged();
    }
    //获取服务器数据
    private void getDataFromServer() {
        OkHttpClient okHttpClient=new OkHttpClient();
        Request request=new Request.Builder().url(WEB_SITE+"?
key="+KEY+"&info="+sendMsg).build();
        Log.i("Aye",WEB_SITE+"?key="+KEY+"&info="+sendMsg);
        //开启异步线程访问网络
        Call call=okHttpClient.newCall(request);
        call.enqueue(new Callback() {
            @Override
            public void onFailure(Call call, IOException e) {
                Log.i("Aye-OkHttp", "未知错误");
            }
            @Override //获取到的信息通过handler传入主线程
            public void onResponse(Call call, Response response) throws
IOException {
                String msg=response.body().string();
                Message resMsg=new Message();
                resMsg.what=1;
                resMsg.obj=msg;
                Log.i("Aye-response",msg);
                handler.sendMessage(resMsg);
            }
        });
    }
    //接收Handler消息
    private Handler handler=new Handler(){
        @Override
        public void dispatchMessage(@NonNull Message msg) {
            super.dispatchMessage(msg);
            switch (msg.what){
                case 1:
                    if (msg.obj!=null){
                        String result=(String)msg.obj;
                        Log.i("Aye-handler",result);
                        paresData(result); //解析消息
                    }
                    break;
            }
        }
    };
    //解析服务器消息
    private void paresData(String content) {
        try {
            JSONObject jsonObject=new JSONObject(content);
            String resContent=jsonObject.getString("text"); //获取返回文本信息
            int code=jsonObject.getInt("code"); //获取返回码
            Log.i("Aye-parse",resContent+","+code);
            updateView(code,resContent); //调用更新界面方法
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    //界面更新
    private void updateView(int code, String resContent) {

```

```

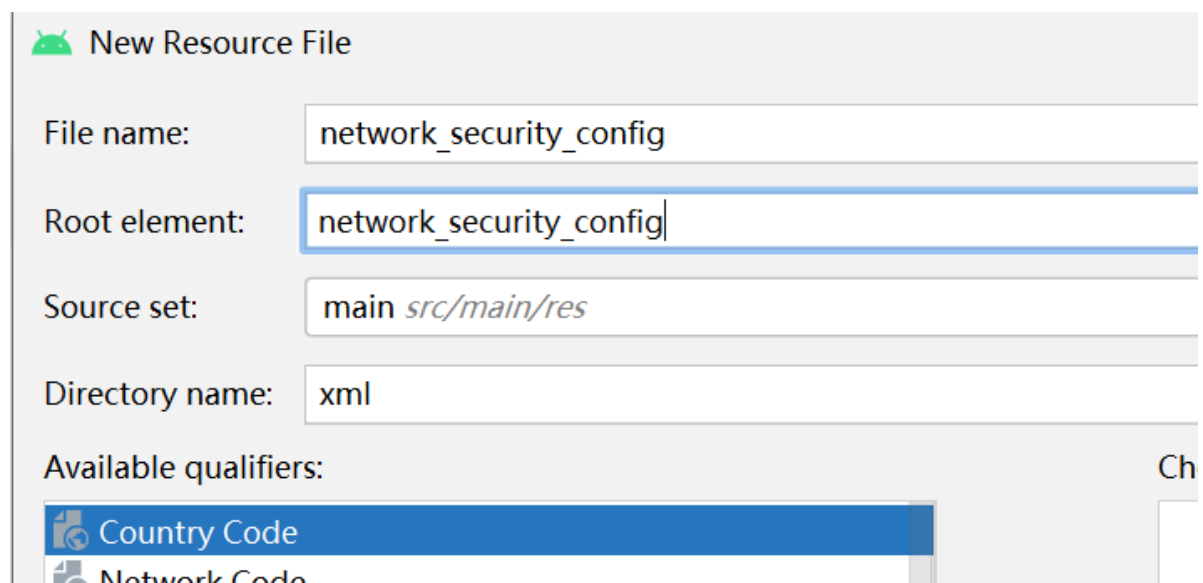
switch (code) {
    case 40001:
        showData("亲爱的，未找到对应的用户信息，请稍后重试。");
    case 40004:
        showData("主人，今天我累了，我要休息了，明天再来找我耍吧");
        break;
    case 40005:
        showData("主人，你说的是外星语吗？");
        break;
    case 40006:
        showData("主人，我今天要去约会哦，暂不接客啦");
        break;
    case 40007:
        showData("主人，明天再和你耍啦，我生病了，呜呜.....");
        break;
    default:
        showData(resContent); //调用该方法显示返回信息
        break;
}
}
}

```

三、关于okhttp3网络框架的onfailure错误

由于Android P全面禁止了非https链接，但是国内的很多网站都是非https。这时候就需要需要设置非安全连接。

在res包下右键创建新的directory，再新建的文件夹里面右键创建xml文件。



编写以下代码

```

<?xml version="1.0" encoding="utf-8"?>
<network-security-config
xmlns:android="http://schemas.android.com/apk/res/android">
    <!--默认允许所有网址使用非安全连接-->
    <base-config cleartextTrafficPermitted="true" />
</network-security-config>

```

在AndroidManifest.xml清单文件中添加一条属性即可。

```
<application  
.....  
    android:networkSecurityConfig="@xml/network_security_config">
```