**实验一**

# 实验项目及基本内容

简单的UI设计实验

1. 安卓开发环境的学习（课下提前准备好）
2. 学习控件的布局及使用
3. 掌握控件的事件响应机制(按钮、输入框、recyclerview）

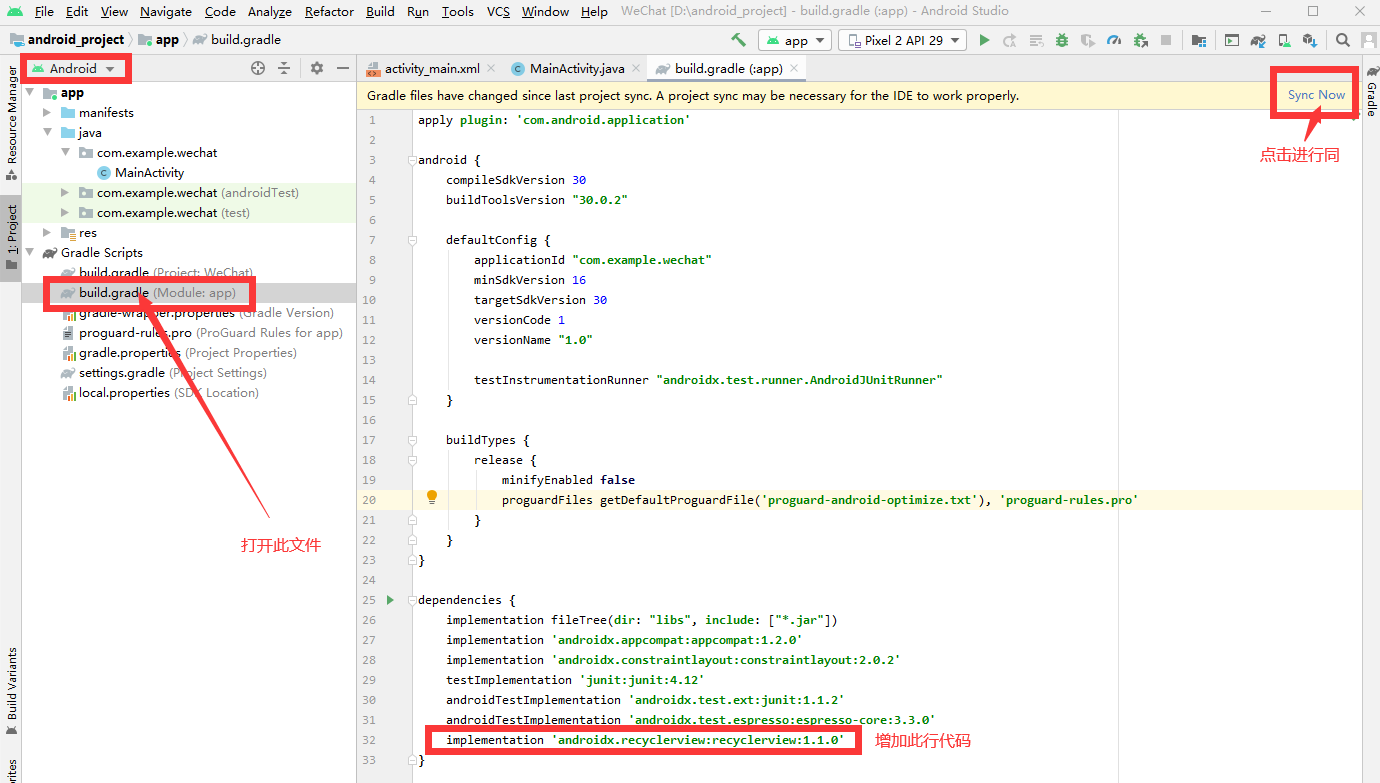
# 类似微信的聊天界面



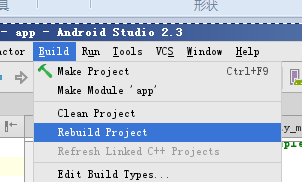
1. 新建一个工程，取名WeChat

在app\build.gradle文件中，dependencies方法最后一行增加如下代码：

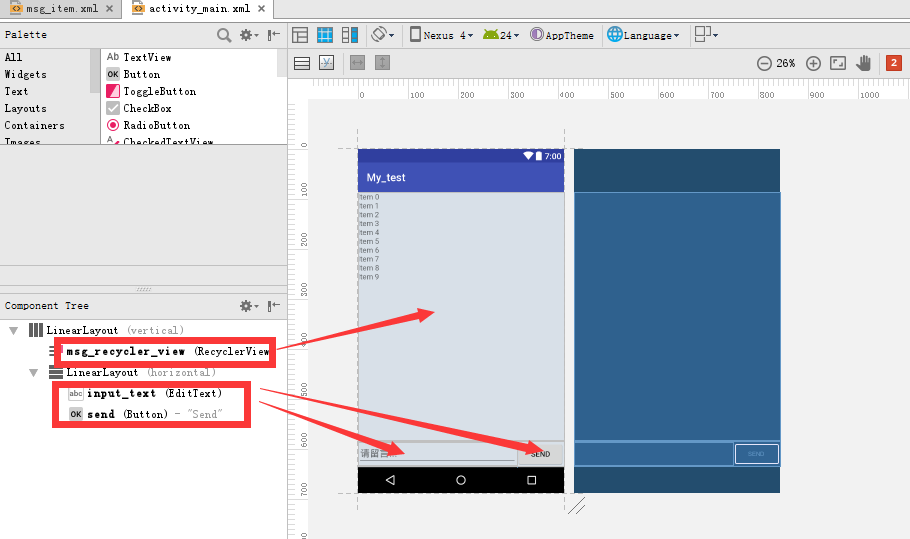
implementation **'androidx.recyclerview:recyclerview:1.1.0'**

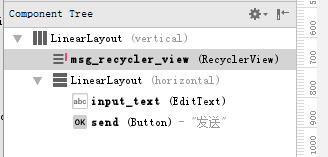


重新rebuild工程，下载支持recyclerview控件的升级包（需要互联网）

****

1. 编辑主界面布局文件activity\_main.xml，上面为recycler\_view控件（msg\_recycler\_view）下面为输入框（input\_text）和发送（send）按钮



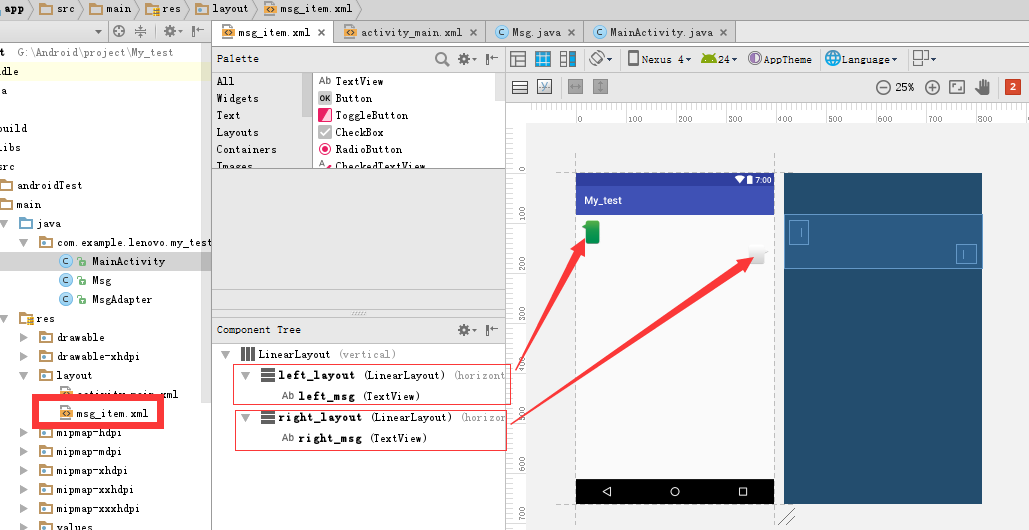


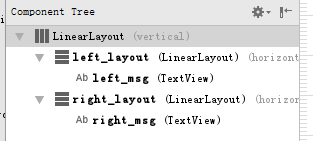
完整代码如下：





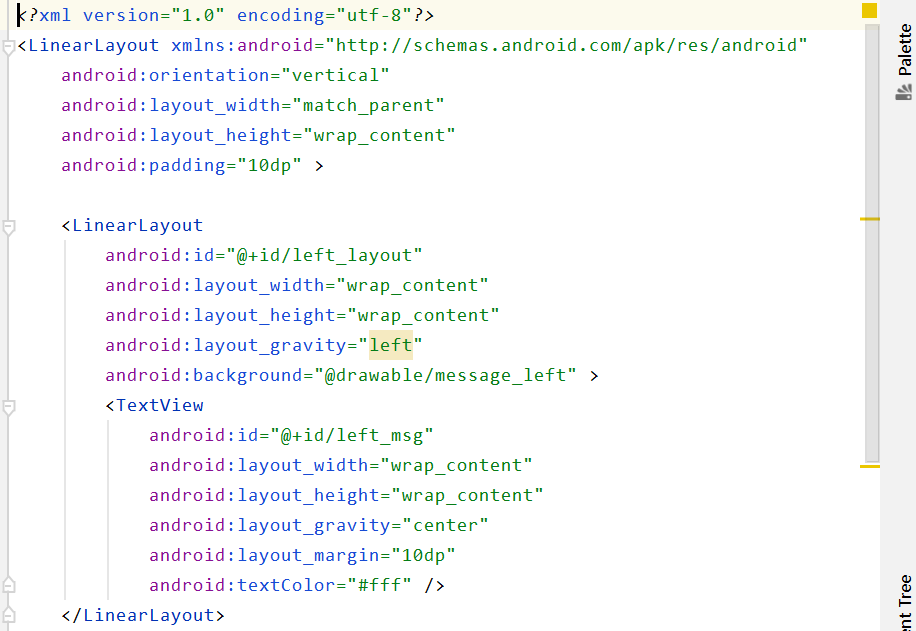
1. 新建一个布局文件msg\_item.xml，显示聊天记录。





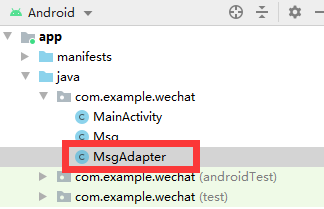
注意：为了区分发送和接收消息，在app\src\main\res\drawable下分别存储两个png文件，每个LinearLayout分别设置背景文件

完整代码如下：





1. 新建MsgAdapter.java文件



将如下代码拷贝到文件

**package** com.example.wechat;  
  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
**import** androidx.recyclerview.widget.RecyclerView;  
**import** java.util.List;  
**public class** MsgAdapter **extends** RecyclerView.Adapter<MsgAdapter.ViewHolder> {  
 **private** List<Msg> **mMsgList**;  
 **static class** ViewHolder **extends** RecyclerView.ViewHolder {  
 LinearLayout **leftLayout**;  
 LinearLayout **rightLayout**;  
 TextView **leftMsg**;  
 TextView **rightMsg**;  
 **public** ViewHolder(View view) {  
 **super**(view);  
 **leftLayout** = (LinearLayout) view.findViewById(R.id.***left\_layout***);  
 **rightLayout** = (LinearLayout) view.findViewById(R.id.***right\_layout***);  
 **leftMsg** = (TextView) view.findViewById(R.id.***left\_msg***);  
 **rightMsg** = (TextView) view.findViewById(R.id.***right\_msg***);  
 }  
 }  
 **public** MsgAdapter(List<Msg> msgList) {  
 **mMsgList** = msgList;  
 }  
 *//生成为每个Item inflater出一个View,返回的是一个ViewHolder。该方法把View直接封装在ViewHolder中，然后我们面向的是ViewHolder这个实例，当然这个ViewHolder需要我们自己去编写* **public** ViewHolder onCreateViewHolder(ViewGroup parent, **int** viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.***msg\_item***, parent, **false**);  
 **return new** ViewHolder(view);  
 }  
 *//适配渲染数据到View* **public void** onBindViewHolder(ViewHolder holder, **int** position) {  
 Msg msg = **mMsgList**.get(position);  
 **if** (msg.getType() == Msg.***TYPE\_RECEIVED***) {  
 *// 如果是收到的消息，则显示左边的消息布局，将右边的消息布局隐藏* holder.**leftLayout**.setVisibility(View.***VISIBLE***);  
 holder.**rightLayout**.setVisibility(View.***GONE***);  
 holder.**leftMsg**.setText(msg.getContent());  
 } **else if**(msg.getType() == Msg.***TYPE\_SENT***) {  
 *// 如果是发出的消息，则显示右边的消息布局，将左边的消息布局隐藏* holder.**rightLayout**.setVisibility(View.***VISIBLE***);  
 holder.**leftLayout**.setVisibility(View.***GONE***);  
 holder.**rightMsg**.setText(msg.getContent());  
 }  
 }  
 **public int** getItemCount() {  
 **return mMsgList**.size();  
 }  
}

1. 创建Msg.java文件，代码如下：
2. **package** com.example.wechat;  
   **public class** Msg {  
    **public static final int *TYPE\_RECEIVED*** = 0;  
    **public static final int *TYPE\_SENT*** = 1;  
    **private** String **content**;  
    **private int type**;  
    **public** Msg(String content, **int** type) {  
    **this**.**content** = content;  
    **this**.**type** = type;  
    }  
    **public** String getContent() {  
    **return content**;  
    }  
    **public int** getType() {  
    **return type**;  
    }  
   }
3. 在MainActivity.java文件中onCreate方法中编写监听程序，接收前台界面的消息，并通过适配器转送给RecyclerView进行显示。

msg：存储从前台界面接收的消息，包括：信息内容、发送（接受）

msgList：存储msg的数组

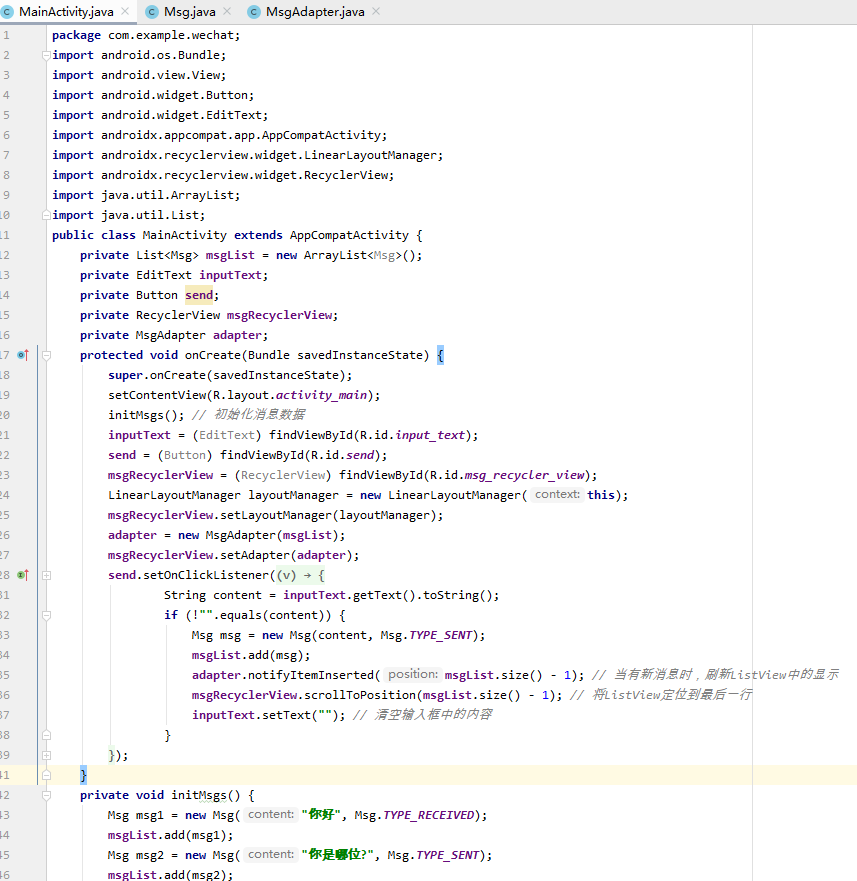
MsgAdapter：适配器，msgRecyclerView控件与msgList的衔接

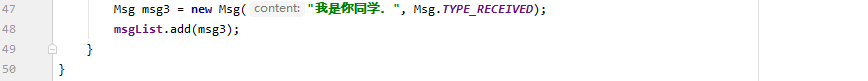
**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 initMsgs(); *// 初始化消息数据* **inputText** = (EditText) findViewById(R.id.***input\_text***);  
 **send** = (Button) findViewById(R.id.***send***);  
 **msgRecyclerView** = (RecyclerView) findViewById(R.id.***msg\_recycler\_view***);  
 LinearLayoutManager layoutManager = **new** LinearLayoutManager(**this**);  
 **msgRecyclerView**.setLayoutManager(layoutManager);  
 **adapter** = **new** MsgAdapter(**msgList**);  
 **msgRecyclerView**.setAdapter(**adapter**);  
 **send**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 String content = **inputText**.getText().toString();  
 **if** (!**""**.equals(content)) {  
 Msg msg = **new** Msg(content, Msg.***TYPE\_SENT***);  
 **msgList**.add(msg);  
 **adapter**.notifyItemInserted(**msgList**.size() - 1); *// 当有新消息时，刷新ListView中的显示* **msgRecyclerView**.scrollToPosition(**msgList**.size() - 1); *// 将ListView定位到最后一行* **inputText**.setText(**""**); *// 清空输入框中的内容* }  
 }  
 });  
}

增加初始化消息方法

**private void** initMsgs() {  
 Msg msg1 = **new** Msg(**"你好"**, Msg.***TYPE\_RECEIVED***);  
 **msgList**.add(msg1);  
 Msg msg2 = **new** Msg(**"你是哪位?"**, Msg.***TYPE\_SENT***);  
 **msgList**.add(msg2);  
 Msg msg3 = **new** Msg(**"我是你同学. "**, Msg.***TYPE\_RECEIVED***);  
 **msgList**.add(msg3);  
}

完整截图：





1. 打包运行程序，观看效果。

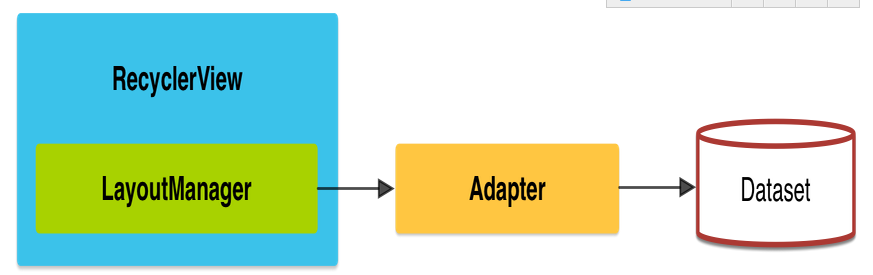
# 附件：RecyclerView简介

RecyclerView 是Android新版本中新添加的一个用来取代ListView的SDK，它的灵活性与可替代性比listview更好。

RecyclerView与ListView原理是类似的：都是仅仅维护少量的View并且可以展示大量的数据集。RecyclerView用以下两种方式简化了数据的展示和处理:

* 使用LayoutManager来确定每一个item的排列方式。
* 为增加和删除项目提供默认的动画效果。

你也可以定义你自己的LayoutManager和添加删除动画，RecyclerView项目结构如下：



* Adapter：使用RecyclerView之前，你需要一个继承自RecyclerView.Adapter的适配器，作用是将数据与每一个item的界面进行绑定。
* LayoutManager：用来确定每一个item如何进行排列摆放，何时展示和隐藏。回收或重用一个View的时候，LayoutManager会向适配器请求新的数据来替换旧的数据，这种机制避免了创建过多的View和频繁的调用findViewById方法（与ListView原理类似）。