

RecyclerView

1. 在 `app/build.gradle` 中添加 `RecyclerView` 依赖

```
compile 'com.android.support:recyclerview-v7:25.3.0'
```

2. 在主活动的布局文件中添加 `RecyclerView` 控件

注意：要加上完整包名 `android.support.v7.widget.RecyclerView`

```
<LinearLayout >
    <android.support.v7.widget.RecyclerView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

3. 创建自定义对象，用于每个 Item 中的控件内容保存

```
public class Fruit {
    private String text;
    private int imageId;
    ...
}
```

4. 创建每个item 对应的布局文件

```
<LinearLayout >
    <ImageView
        />
    <TextView
        />

</LinearLayout>
```

5. 创建自定义Adapter，继承 `RecyclerView.Adapter<? extends RecyclerView.ViewHolder>`

注意：需要传入 `RecyclerView.ViewHolder` 子类 作为泛型，所以要创建内部类 `ViewHolder`

重写三个方法

- `ViewHolder onCreateViewHolder(final ViewGroup parent, int viewType)`

创建 `ViewHolder` 对象，需要加载每个item对应的 布局文件，返回View 对象，再通过该 `View`

对象中的控件，创建 `ViewHolder` 对象

创建 `ViewHolder` 对象之后，可以设置 其中的每个控件的响应事件,这是 `ListView` 无法做到的

```
viewHolder.imageView.setOnClickListener(new View.OnClickListener()
```

- `void onBindViewHolder(FruitAdapter.ViewHolder holder, int position)`
为每个 `ViewHolder` 中的控件绑定具体内容
- `int getItemCount()`
返回当前列表项的总数

创建内部类继承 `RecyclerView.ViewHolder`

在内部类中，要获取 `onCreateViewHolder()` 中 加载的 每个item的布局 `View` 中的控件对象

6. 主活动加载

- 创建自定义Adapter对象，创建数组/ArrayList<>，初始化数据
- 获取RecyclerView 控件对象，创建LinearLayoutManager 对象，用于给 `RecyclerView` 设置 `RecyclerView` 的布局方式(`setLayoutManager()`)
- 给 `RecyclerView` 设置 自定义 adapter 对象

7. 方法说明

- `int position = viewHolder.getAdapterPosition();`
//获取当前holder 所在 recyclerView 中的位置
- `Context context= view.getContext();`
获取当前View对象 所在的上下文 (Context)
- `View v =`
`LayoutInflater.from(parent.getContext()).inflate(R.layout.fruit_item,parent,false);`
加载布局,返回View对象
- `holder.leftLayout.setVisibility(View.VISIBLE);`
设置Layout布局控件为 可见(View.VISIBLE)/ 不可见(View.GONE);
- `adapter.notifyItemInserted(msgList.size() - 2);` 通知列表插入了新的 内容，传入新插入的内容在 容器中的 位置, 列表将更新该位置的数据 为 容器中该位置的内容
- `recyclerView.scrollToPosition(msgList.size() - 1);`
将显示的数据 定位到最后一行

源码

自定义 `Adapter` 类

```

public class FruitAdapter extends RecyclerView.Adapter<FruitAdapter.ViewHolder> {

    ArrayList<Fruit> fruits;
    public FruitAdapter(ArrayList<Fruit> a){
        this.fruits = a;
    }
    //内部类
    static class ViewHolder extends RecyclerView.ViewHolder{
        View fruitView; // 保存子项
        ImageView imageView;
        TextView textView;
        public ViewHolder(View v){
            super(v);
            fruitView = v;
            imageView = (ImageView)v.findViewById(R.id.image);
            textView = (TextView)v.findViewById(R.id.text_view);

        }
    }
    @Override
    public FruitAdapter.ViewHolder onCreateViewHolder(final ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.fruit_item,parent,false);
        final ViewHolder viewHolder = new ViewHolder(v);
        // viewHolder.fruitView 实际上就是 v
        viewHolder.fruitView.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                int position = viewHolder.getAdapterPosition();
                Fruit fruit = fruits.get(position);
                Toast.makeText(v.getContext(),"您点击的是第"+position+"个子项",Toast.LENGTH_SHORT).show();
            }
        });
        viewHolder.imageView.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                //获取当前holder 所在 recyclerView 中的位置
                int position = viewHolder.getAdapterPosition();
                Fruit fruit = fruits.get(position);
                Toast.makeText(v.getContext(),"您点击的是第"+position+"张图片",Toast.LENGTH_SHORT).show();
            }
        });
        return viewHolder;
    }

    @Override

```

```
    public void onBindViewHolder(FruitAdapter.ViewHolder holder, int position) {
        Fruit fruit = fruits.get(position);
        holder.imageView.setImageResource(fruit.getImageId());
        holder.textView.setText(fruit.getText());
    }

    @Override
    public int getItemCount() {
        return fruits.size();
    }
}
```

主活动

```
initFruits();
FruitAdapter adapter = new FruitAdapter(list);
LinearLayoutManager manager = new LinearLayoutManager(this);
RecyclerView recyclerView = (RecyclerView)findViewById(R.id.list_view);
recyclerView.setLayoutManager(manager);
recyclerView.setAdapter(adapter);
```