

效果



代码

实体类必须实现SectionMultiEntity，在设置数据的时候，需要给每一个数据设置itemType，泛型中实体类为主体item真正数据源 ``java public class SectionMultipleItem extends SectionMultiEntity

implements MultiItemEntity {

```
public static final int TEXT = 1;
public static final int IMG_TEXT = 2;
private int itemType;          // 附加字段方便处理所需逻辑
private boolean isMore;        // 附加字段方便处理所需逻辑
private Video video;           // 主体数据实体类型

// 创建section 数据
public SectionMultipleItem(boolean isHeader, String header, boolean isMore) {
    super(isHeader, header);
    this.isMore = isMore;
}

// 创建主体item数据
public SectionMultipleItem(int itemType, Video video) {
    super(video);
    this.video = video;
    this.itemType = itemType;
}

public boolean isMore() {
    return isMore;
}

public void setMore(boolean more) {
    isMore = more;
}

public Video getVideo() {
    return video;
}

public void setVideo(Video video) {
    this.video = video;
}

@Override
public int getItemType() {
    return itemType;
}
}
```

在构造里面`addItemType`绑定type和layout的关系，adapter构造需要传入head的布局id，在convert方法里面加载item数据，在convertHead方法里面加载head数据java public class SectionMultipleItemAdapter extends BaseSectionMultiItemQuickAdapter<SectionMultipleItem, BaseViewHolder> { /** * init SectionMultipleItemAdapter * 1. add your header resource layout * 2. add some kind of items ** @param sectionHeadResId The section head layout id for each item * @param data A new list is created out of this one to avoid mutable list */ public SectionMultipleItemAdapter(int sectionHeadResId, List data) { super(sectionHeadResId, data); addItemType(SectionMultipleItem.TEXT, R.layout.item_text_view); addItemType(SectionMultipleItem.IMG_TEXT, R.layout.item_img_text_view); }

```
@Override
protected void convertHead(BaseViewHolder helper, final SectionMultipleItem item) {
    // deal with header viewHolder
    helper.setText(R.id.header, item.header);
}
```

```

        helper.setVisible(R.id.more, item.isMore());
        helper.setOnClickListener(R.id.more);
    }

    @Override
    protected void convert(BaseViewHolder helper, SectionMultipleItem item) {
        // deal with multiple type items viewHolder
        helper.setOnClickListener(R.id.card_view);
        switch (helper.getItemViewType()) {
            case MultipleItem.TEXT:
                helper.setText(R.id.tv, item.getVideo().getName());
                break;
            case MultipleItem.IMG_TEXT:
                switch (helper.getLayoutPosition() % 2) {
                    case 0:
                        helper.setImageResource(R.id.iv, R.mipmap.animation_img1);
                        break;
                    case 1:
                        helper.setImageResource(R.id.iv, R.mipmap.animation_img2);
                        break;
                }
                break;
        }
    }
}

```

} section_multiple类型需要继承特定实体类，满足特定规则`SectionMultiEntity`， public static List getSectionMultiData() { List list = new ArrayList<>();
Video video = new Video(HTTPS_AVATARS1_GITHUBUSERCONTENT_COM_LINK, CYM_CHAD);

```

// 以下为列表的一个section item数据
// 每个item包含一个section + n 个 video 数据。 video 为主体item所需数据类型
// add section data
list.add(new SectionMultipleItem(true, "Section 1", true));
// add multiple type item data ---start---
list.add(new SectionMultipleItem(SectionMultipleItem.TEXT, new Video(HTTPS_AVATARS1_GITHUBUSERCONTENT_COM_LINK, "video_id_0")))
list.add(new SectionMultipleItem(SectionMultipleItem.TEXT, new Video(HTTPS_AVATARS1_GITHUBUSERCONTENT_COM_LINK, "video_id_1")))
list.add(new SectionMultipleItem(SectionMultipleItem.IMG_TEXT, new Video(HTTPS_AVATARS1_GITHUBUSERCONTENT_COM_LINK, "video_id_2")))
// ---end---

list.add(new SectionMultipleItem(true, "Section 2", false));
list.add(new SectionMultipleItem(SectionMultipleItem.IMG_TEXT, video));
list.add(new SectionMultipleItem(SectionMultipleItem.IMG_TEXT, video));
list.add(new SectionMultipleItem(SectionMultipleItem.TEXT, video));
list.add(new SectionMultipleItem(true, "Section 3", false));
list.add(new SectionMultipleItem(SectionMultipleItem.IMG_TEXT, video));
list.add(new SectionMultipleItem(SectionMultipleItem.TEXT, video));
return list;
}

```

...

数据回调，可根据adapter数据源获取主体item实体类 `` sectionAdapter.setOnItemChildClickListener(new
BaseQuickAdapter.OnItemChildClickListener() { @Override public void onItemChildClick(BaseQuickAdapter adapter, View view, int position) {
SectionMultipleItem item = (SectionMultipleItem) adapter.getData().get(position); switch (view.getId()) { case R.id.card_view: // 获取主体item相应数
据给后期使用 if (item.getVideo() != null) { Toast.makeText(SectionMultipleItemUseActivity.this, item.getVideo().getName(),
Toast.LENGTH_LONG).show(); } break; default: Toast.makeText(SectionMultipleItemUseActivity.this, "OnItemChildClickListener " + position,
Toast.LENGTH_LONG).show(); break;

```

        }
    }
});

```

...