DataBinding

基于 BaseQuickAdapter 的用法：

代码如下：

```
public class DataBindingAdapter extends BaseQuickAdapter<Movie, BaseViewHolder> {
    private MoviePresenter mPresenter = new MoviePresenter();

    public DataBindingAdapter() {
        super(R.layout.item_movie);
    }

    /**
     * 当 ViewHolder 创建完毕以后，会执行此回掉
     * 可以在这里做任何你想做的事情
     */
    @Override
    protected void onItemViewHolderCreated(@NotNull BaseViewHolder viewHolder, int viewType) {
        // 绑定 view
        DataBindingUtil.bind(viewHolder.itemView);
    }

    @Override
    protected void convert(@NotNull BaseViewHolder helper, @NotNull Movie item) {
        if (item == null) {
            return;
        }

        // 获取 Binding
        ItemMovieBinding binding = helper.getBinding();
        if (binding != null) {
            // 设置数据
            binding.setMovie(item);
            binding.setPresenter(mPresenter);
            binding.executePendingBindings();
        }
    }
}
```

基于 **BaseProviderMultiAdapter** 的用法

由于使用了`Provider`，所以相关事务并不在`Adapter`里处理。需要在`Provider`中对数据进行绑定。

`Adapter`代码如下（无需特殊处理，和普通的`BaseProviderMultiAdapter`写法一样）：

```
public class ProviderMultiAdapter extends BaseProviderMultiAdapter<ProviderMultiEntity> {

    public ProviderMultiAdapter() {
        super();
        addItemProvider(new ImgItemProvider());
        addItemProvider(new TextImgItemProvider());
        addItemProvider(new TextItemProvider());
    }

    /**
     * 自行根据数据、位置等内容，返回 item 类型
     */
    @Override
    protected int getItemType(@NotNull List<? extends ProviderMultiEntity> data, int position) {
        switch (position % 3) {
            case 0:
                return ProviderMultiEntity.IMG;
            case 1:
                return ProviderMultiEntity.TEXT;
            case 2:
                return ProviderMultiEntity.IMG_TEXT;
            default:
                break;
        }
        return 0;
    }
}
```

`Provider` 代码如下：

```java
public class ImgItemProvider extends BaseItemProvider<ProviderMultiEntity> {

    @Override
    public int getItemViewType() {
        return ProviderMultiEntity.IMG;
    }

    @Override
    public int getLayoutId() {
        return R.layout.item_image_view;
    }

    /**
     * 当 ViewHolder 创建完毕以后，会执行此回掉
     * 可以在这里做任何你想做的事情
     */
    @Override
    public void onViewHolderCreated(@NotNull BaseViewHolder viewHolder) {
        // 绑定View
        DataBindingUtil.bind(viewHolder.itemView);
    }

    @Override
    public void convert(@NotNull BaseViewHolder helper, @NotNull ProviderMultiEntity data) {
        Movie movie = data.getMovie();
        // 获取 Binding
        ItemMovieBinding binding = helper.getBinding();
        if (binding != null) {
            binding.setMovie(movie);
            binding.executePendingBindings();
        }
    }
}
```