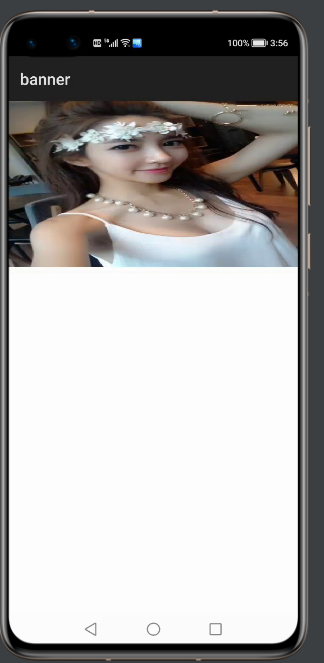
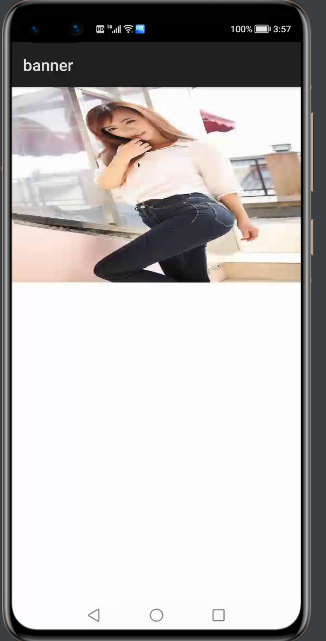
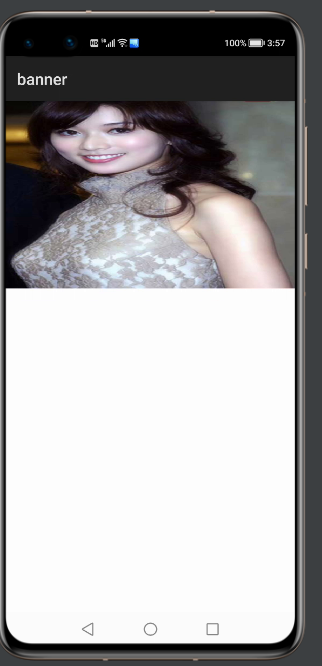
1. **BannerPageSlider功能介绍**
   1. **组件介绍**

应用开发过程，用鸿蒙提供的“PageSlider”结合“EventHandler”实现轮播图效果,代码量过大。而BannerPageSlider组件可以更快速实现一个简单的轮播图功能

* 1. **phone模拟器上运行效果**

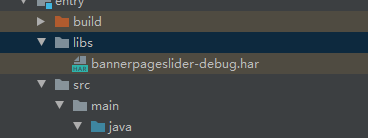




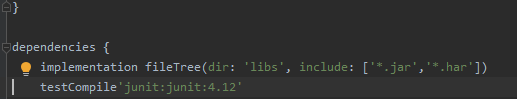


1. **BannerPageSlider使用方法**
   1. **为应用添加bannerpageslider-debug.har包依赖**

在应用模块中调用HAR，常用的添加依赖的方式包括如下两种。

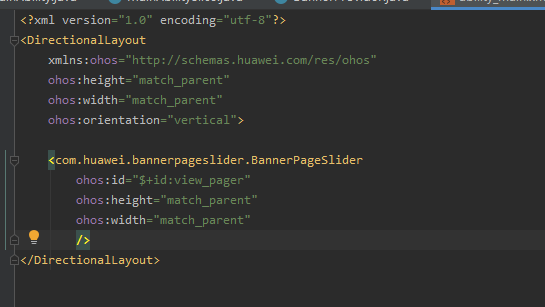
* 方式一：依赖本地HAR,将**bannerpageslider-debug.har**复制到entry\libs目录下即可（由于build.gradle已经依赖的libs目录下的\*.har，因此不需要在做修改）。
* 

查看工程目录中build.gradle下的\*.har是存在



以上操作无误之后就可以进行编码了！

1. **BannerPageSlider开发实现**
   1. **主页面的布局文件**



* 1. **MainAbilitySlice代码如下**

**// 定义数据源  
private List<Image> data;**

**// 定义BannerPageSilder组件对象  
private BannerPageSlider autoPlayViewPage;**

**// 定义一个适配器  
private BannerProvider bannerProvider;  
@Override  
public void onStart(Intent intent) {  
 super.onStart(intent);  
 super.setUIContent(ResourceTable.*Layout\_ability\_main*);**

**// 实例化数据源  
 data = new ArrayList<>();  
 Image image1 = new Image(this);  
 Image image2 = new Image(this);  
 Image image3 = new Image(this);**

**// 设置image的宽高，并且图片充满整个image  
image1.setLayoutConfig(new ComponentContainer.LayoutConfig(ComponentContainer.LayoutConfig.*MATCH\_PARENT*, ComponentContainer.LayoutConfig.*MATCH\_CONTENT*));**

**// 设置图片充满image  
 image1.setScaleMode(Image.ScaleMode.*STRETCH*);  
 image2.setLayoutConfig(new ComponentContainer.LayoutConfig(ComponentContainer.LayoutConfig.*MATCH\_PARENT*, ComponentContainer.LayoutConfig.*MATCH\_CONTENT*));  
 image2.setScaleMode(Image.ScaleMode.*STRETCH*);  
 image3.setLayoutConfig(new ComponentContainer.LayoutConfig(ComponentContainer.LayoutConfig.*MATCH\_PARENT*, ComponentContainer.LayoutConfig.*MATCH\_CONTENT*));  
 image3.setScaleMode(Image.ScaleMode.*STRETCH*);  
 // 给图片设置资源  
 image1.setPixelMap(ResourceTable.*Media\_ku*);  
 image2.setPixelMap(ResourceTable.*Media\_ku2*);  
 image3.setPixelMap(ResourceTable.*Media\_ku3*);  
 // 给list增加数据  
 data.add(image1);  
 data.add(image2);  
 data.add(image3);  
 // 实例化适配器  
 bannerProvider = new BannerProvider(this,data);  
 autoPlayViewPage = (BannerPageSlider) findComponentById(ResourceTable.*Id\_view\_pager*);  
 autoPlayViewPage.setProvider(bannerProvider);  
 // 以下两个方法不是必须的，因为有默认值  
 autoPlayViewPage.setDirection(BannerPageSlider.Direction.*RIGHT*);// 设置播放方向  
 autoPlayViewPage.setCurrentPage(200); // 设置每个Item展示的时间  
 autoPlayViewPage.start(); // 开始轮播  
}**

* 1. **BannerProvider 适配器代码如下**

**public class BannerProvider extends PageSliderProvider {  
 private List<Image> listData;  
 private Context mContext;  
  
  
 public BannerProvider(Context context, List<Image> listData) {  
 this.mContext = context;  
 this.listData = listData;  
 }  
  
 @Override  
 public int getCount() {  
 return listData.size();  
 }  
  
  
 @Override  
public Object createPageInContainer(ComponentContainer componentContainer, int i) {  
 i = i % listData.size();  
 Object xxoo = listData.get(i);  
 componentContainer.addComponent((Component) xxoo);  
 return xxoo;  
 }  
  
 @Override  
 public void destroyPageFromContainer(ComponentContainer componentContainer, int i, Object o) {  
 componentContainer.removeComponent(listData.get(i));  
 }  
  
 @Override  
 public boolean isPageMatchToObject(Component component, Object o) {  
 return component == o;  
 }  
}**