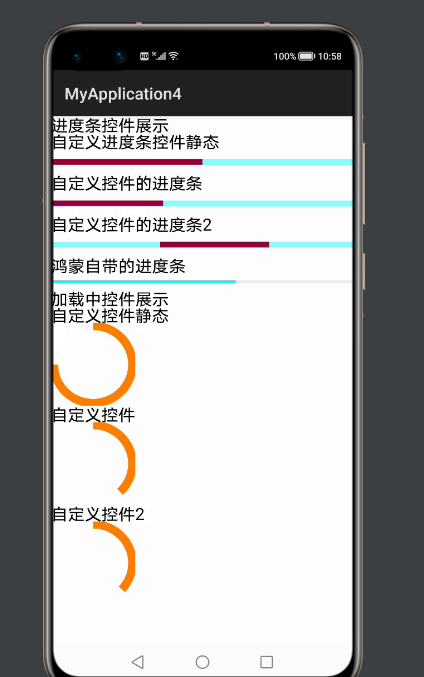
1. **MaterialProgressBarohos功能介绍**
   1. **组件介绍**

基于鸿蒙系统进行的自定义控件的进度条以及加载中的控件展示，使开发者在开发过程中有一个更加方便的开发过程，以及可以根据自己的要求去定义宽度，高度，进度条的粗细，进度的加载快慢，以及各种颜色的设置。

* 1. **phone模拟器上运行效果**

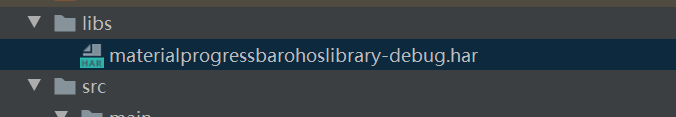


1. **MaterialProgressBarohos使用方法**
   1. **为应用添加MaterialProgressBarohoslibrary-debug.har包依赖**

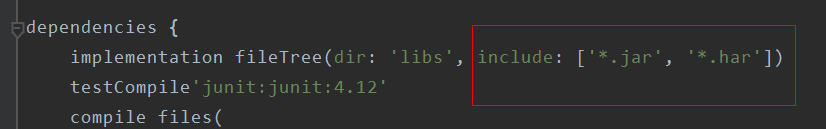
在应用模块中调用HAR，常用的添加依赖的方式包括如下两种。

* 方式一：依赖本地HAR

将**MaterialProgressBarohoslibrary-debug.har**复制到entry\libs目录下即可（由于build.gradle中已经依赖的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. **ActiveOhos开发实现**
   1. **主页面的布局文件**

定义六个按钮分别展示各种进度条

<?xml version="1.0" encoding="utf-8"?>  
<DirectionalLayout  
 xmlns:ohos="http://schemas.huawei.com/res/ohos"  
 ohos:height="match\_parent"  
 ohos:width="match\_parent"  
 ohos:orientation="vertical">  
  
  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="进度条控件展示"/>  
  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="自定义进度条控件静态"/>  
 <Button  
 ohos:height="30vp"  
 ohos:width="match\_parent"  
 ohos:background\_element="$graphic:background\_ability\_main"  
 ohos:id="$+id:button1"  
 />  
  
 <Text  
 ohos:height="20vp"  
 ohos:text\_size="20fp"  
 ohos:width="match\_parent"  
 ohos:text="自定义控件的进度条"/>  
 <Button  
 ohos:height="30vp"  
 ohos:width="match\_parent"  
 ohos:background\_element="$graphic:background\_ability\_main"  
 ohos:id="$+id:button2"  
 />  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="自定义控件的进度条2"/>  
 <Button  
 ohos:height="30vp"  
 ohos:width="match\_parent"  
 ohos:background\_element="$graphic:background\_ability\_main"  
 ohos:id="$+id:button3"  
 />  
  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="鸿蒙自带的进度条"/>  
 <ProgressBar  
 ohos:height="20vp"  
 ohos:width="match\_parent"  
 ohos:progress="0"  
 ohos:max="100"  
 ohos:id="$+id:progressBar"/>  
  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="加载中控件展示"/>  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="自定义控件静态"/>  
  
 <Button  
 ohos:height="100vp"  
 ohos:width="100vp"  
 ohos:id="$+id:button4"/>  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="自定义控件"/>  
 <Button  
 ohos:height="100vp"  
 ohos:width="100vp"  
 ohos:id="$+id:button5"  
 />  
 <Text  
 ohos:height="20vp"  
 ohos:width="600vp"  
 ohos:text\_size="20fp"  
 ohos:text="自定义控件2"/>  
 <Button  
 ohos:height="100vp"  
 ohos:width="100vp"  
 ohos:id="$+id:button6"  
 />  
  
</DirectionalLayout>

* 1. **例子代码如下**

组件一共有两种，一种是线性进度条，一种是圆形进度条

***MainAbilitySlice***

import com.example.materialprogressbarohoslibrary.CircleProgress;  
import com.example.materialprogressbarohoslibrary.CircleProgressDrawTask;  
import com.example.materialprogressbarohoslibrary.LineProgress;  
import com.example.materialprogressbarohoslibrary.LineProgressDrawTask;  
import com.example.myapplication4.ResourceTable;  
import ohos.aafwk.ability.AbilitySlice;  
import ohos.aafwk.content.Intent;  
import ohos.agp.components.Component;  
import ohos.agp.components.ProgressBar;  
import ohos.eventhandler.EventHandler;  
import ohos.eventhandler.EventRunner;  
  
  
public class MainAbilitySlice extends AbilitySlice {  
  
 public EventHandler handler = new EventHandler(EventRunner.*create*(true));  
  
 @Override  
 public void onStart(Intent intent) {  
 super.onStart(intent);  
 super.setUIContent(ResourceTable.*Layout\_ability\_main*);  
 Component button1 = findComponentById(ResourceTable.*Id\_button1*);  
 Component button2 = findComponentById(ResourceTable.*Id\_button2*);  
 Component button3 = findComponentById(ResourceTable.*Id\_button3*);  
 Component button4 = findComponentById(ResourceTable.*Id\_button4*);  
 Component button5 = findComponentById(ResourceTable.*Id\_button5*);  
 Component button6 = findComponentById(ResourceTable.*Id\_button6*);  
 ProgressBar progressBar = (ProgressBar) findComponentById(ResourceTable.*Id\_progressBar*);  
  
 LineProgressDrawTask lineProgressDrawTask = new LineProgressDrawTask(button1);  
 LineProgressDrawTask lineProgressDrawTask2 = new LineProgressDrawTask(button2);  
 LineProgressDrawTask lineProgressDrawTask3 = new LineProgressDrawTask(button3);  
 lineProgressDrawTask.setCurrentValue(50);  
  
 LineProgress lineProgress = new LineProgress(this);  
 lineProgress.setStartNum(2);  
 lineProgress.SetMoreAnimator(lineProgressDrawTask2,100,200);  
 lineProgress.SetMoreAnimator(lineProgressDrawTask3,100,200);  
 lineProgress.SetMoreAnimator2(lineProgressDrawTask3,100,200);  
  
 CircleProgressDrawTask mDrawTask = new CircleProgressDrawTask(button4);  
 mDrawTask.setMaxValue(100);  
 mDrawTask.setCurrentValue(75);  
 CircleProgressDrawTask mDrawTask2 = new CircleProgressDrawTask(button5);  
 mDrawTask.setMaxValue(100);  
 CircleProgressDrawTask mDrawTask3 = new CircleProgressDrawTask(button6);  
 mDrawTask.setMaxValue(100);  
  
 CircleProgress circleProgress = new CircleProgress(this);  
 circleProgress.SetMoreAnimator(mDrawTask2,100,200);  
 circleProgress.SetMoreAnimator(mDrawTask3,100,200);  
 circleProgress.SetMoreAnimator2(mDrawTask3,100,200);  
  
 handler.postTask(new Runnable() {  
 int i = 0;  
 @Override  
 public void run() {  
 i = i + 1;  
 getUITaskDispatcher().asyncDispatch(new Runnable(){  
 @Override  
 public void run() {  
 if( i <= mDrawTask.getMaxValue()) {  
 progressBar.setProgressValue(i);  
  
 } else {  
 progressBar.setProgressValue(0);  
 i = 0;  
 }  
 }  
 });  
 handler.postTask(this, 100);  
 }  
 }, 200);  
 }  
  
 @Override  
 public void onActive() {  
 super.onActive();  
 }  
  
 @Override  
 public void onForeground(Intent intent) {  
 super.onForeground(intent);  
 }  
}