《鸿蒙应用开发实战》

——修订说明

**目录**

[2.1.2 安装及配置DevEco Studio 2](#_Toc75796249)

[2.2 Hello World 7](#_Toc75796250)

[2.2 Hello World （第27页 第四段） 15](#_Toc75796251)

[3.6 任务6：将主页面中选择器的值传递到训练页面 16](#_Toc75796252)

[3.23 任务23：在第2个训练报告页面中显示除心率曲线之外的所有内容 17](#_Toc75796253)

[3.26 任务26：在第三个训练报告页面中显示除活动分布图之外的所有内容 19](#_Toc75796254)

[路由修复 39](#_Toc75796255)

2.1.2 安装及配置DevEco Studio

**存在问题**

官网更新，DevEco Studio的版本更新，配置环境和新键项目的UI界面变化。

**修订内容**

随着DevEco Studio的不断更新，其官网也随之变化，DevEco Studio的版本与下载链接的位置有所变化。DevEco Studio官网页面如图1所示。

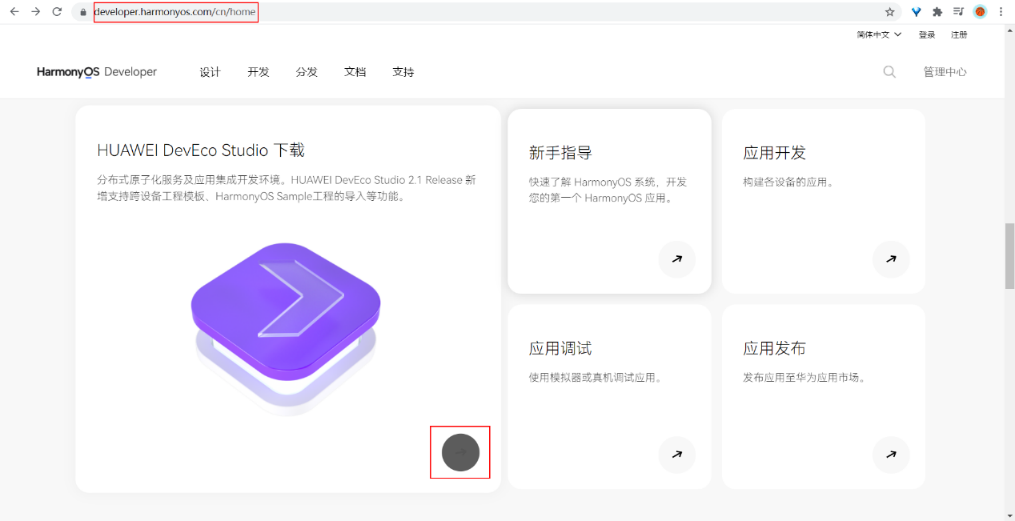


图1 DecEco Studio的官网

单击进入后，在新打开的页面中，会看到两个版本的DevEco Studio的版本有两个，分别为2.1 Release和2.2 Beta1版本。其中Release版本提供更为稳定的功能特性，Bate版本提供更新鲜的功能特性，作为鸿蒙系统的学习者，我们选择更为稳定的Release版本就可以了。DeveEco Studio的官网下载页面如图2所示。

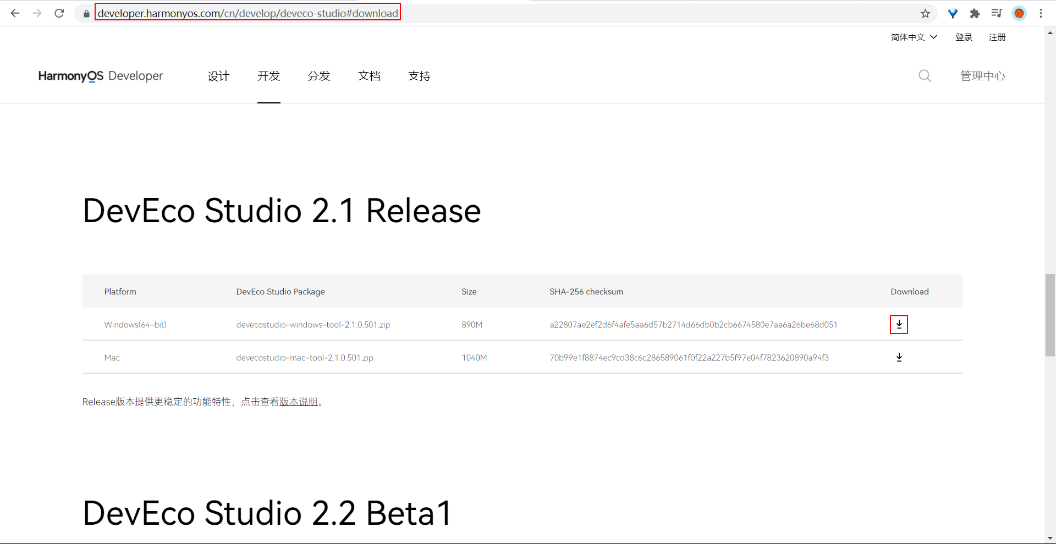


图2 DeveEco Studio的官网下载页面

其它操作和书上一样，安装完成及以后的操作如下。

安装完成后，在新打开的窗口中显示DevEco Studio已经被安装在了你的电脑上，如同3所示。选中Reboot now复选框，然后单击Finish按钮进入下一步，电脑会进行重启。

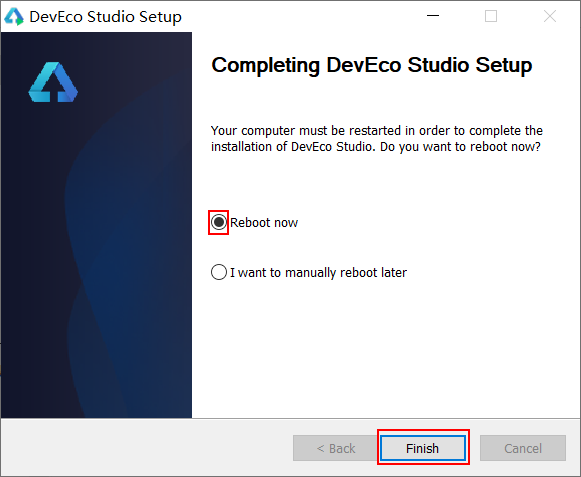


图3 完成安装DevEco Studio

电脑重启完毕后双击桌面上DevEco Studio 2.1.0.501 x64图标，需要确认已经阅读并接受了用户许可协议中的条款和条件，如图4所示。单击Agree按钮进入下一步。

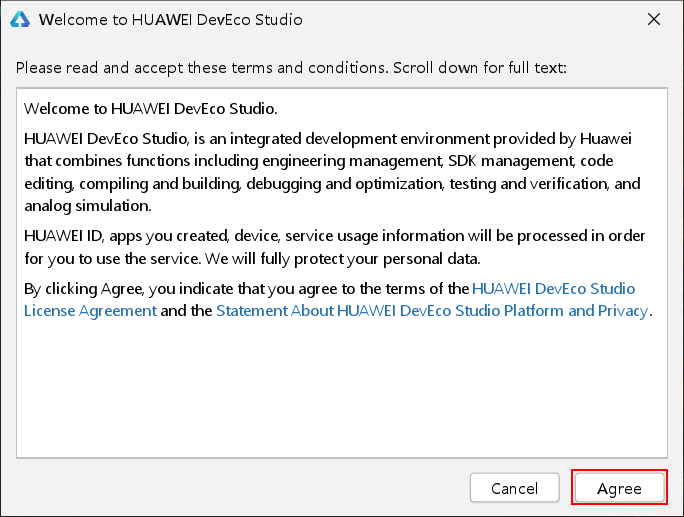


图4 DevEco Studio用户许可协议

在新打开的窗口中点击Configure按钮，然后再展开的下拉菜单中单击Settings，如图5所示。

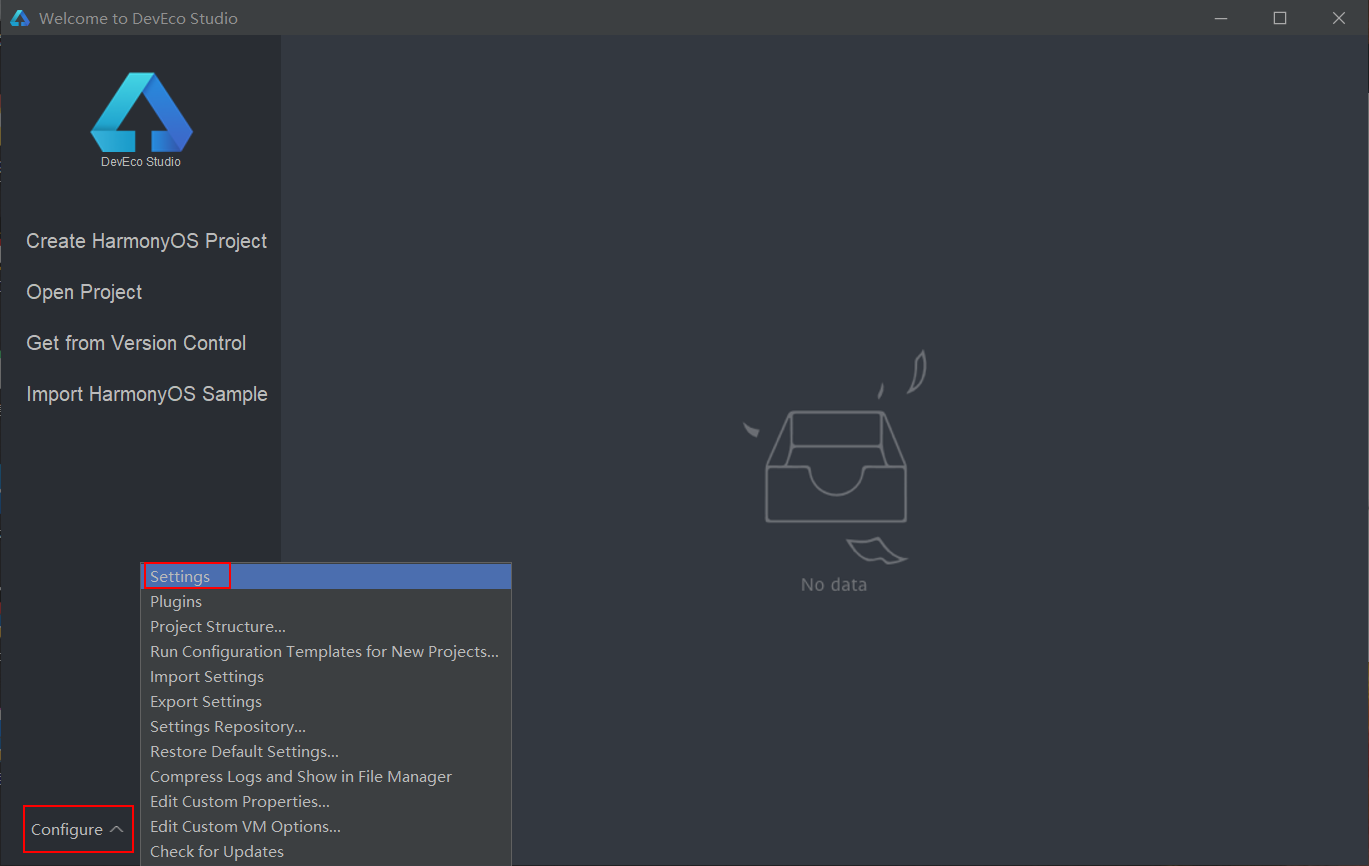


图5 配置DevEco Studio

在新打开的窗口中，单击窗口左侧的Harmony OS SDK，然后单击窗口右侧的SDK Platforms选项卡，将SDK（API Version 5）和SDK（API Version 4）中的Js和Java复选框选中，如图6所示。

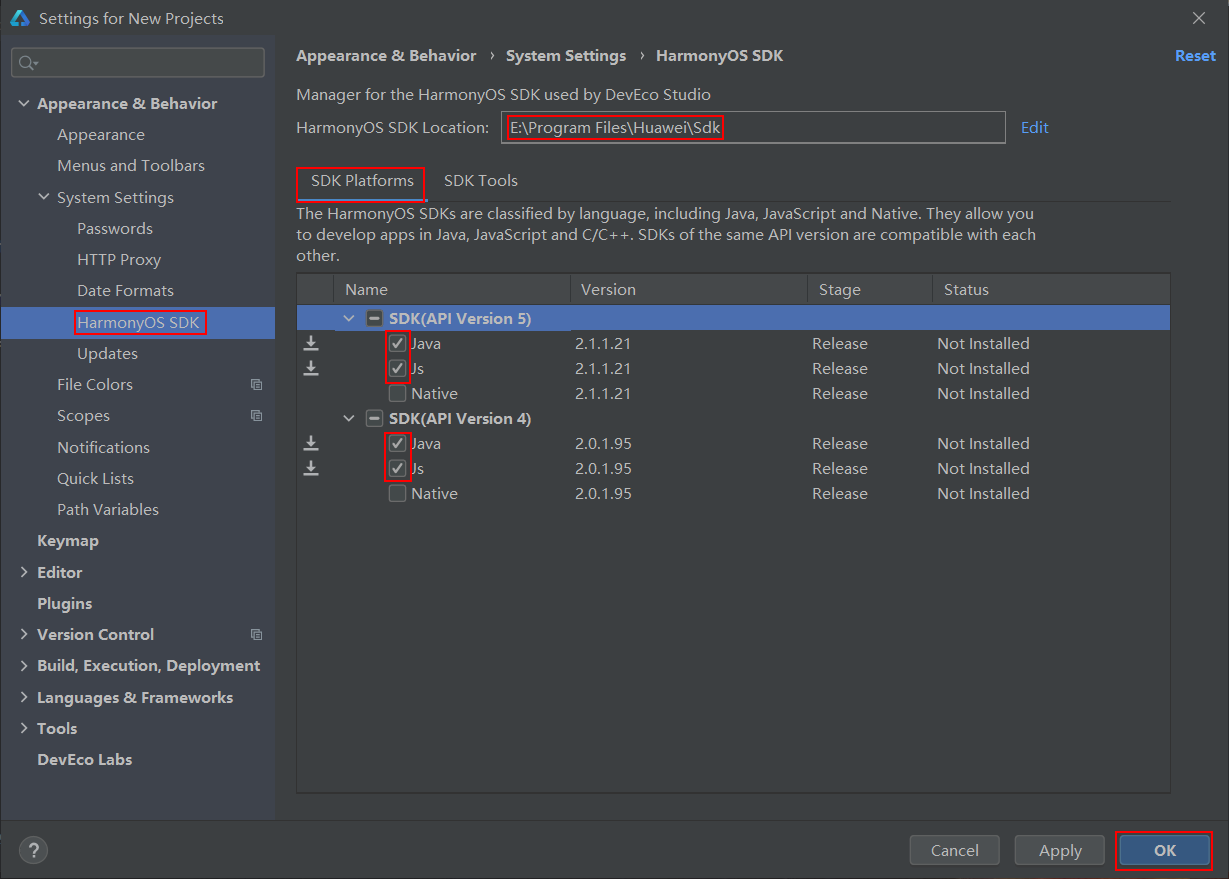


图6 配置安装SDK中的Js组件

单击窗口右侧的SDK Tools选项卡，并选中Previewer和Toolchains复选框，如图7所示。单击OK按钮进入下一步。

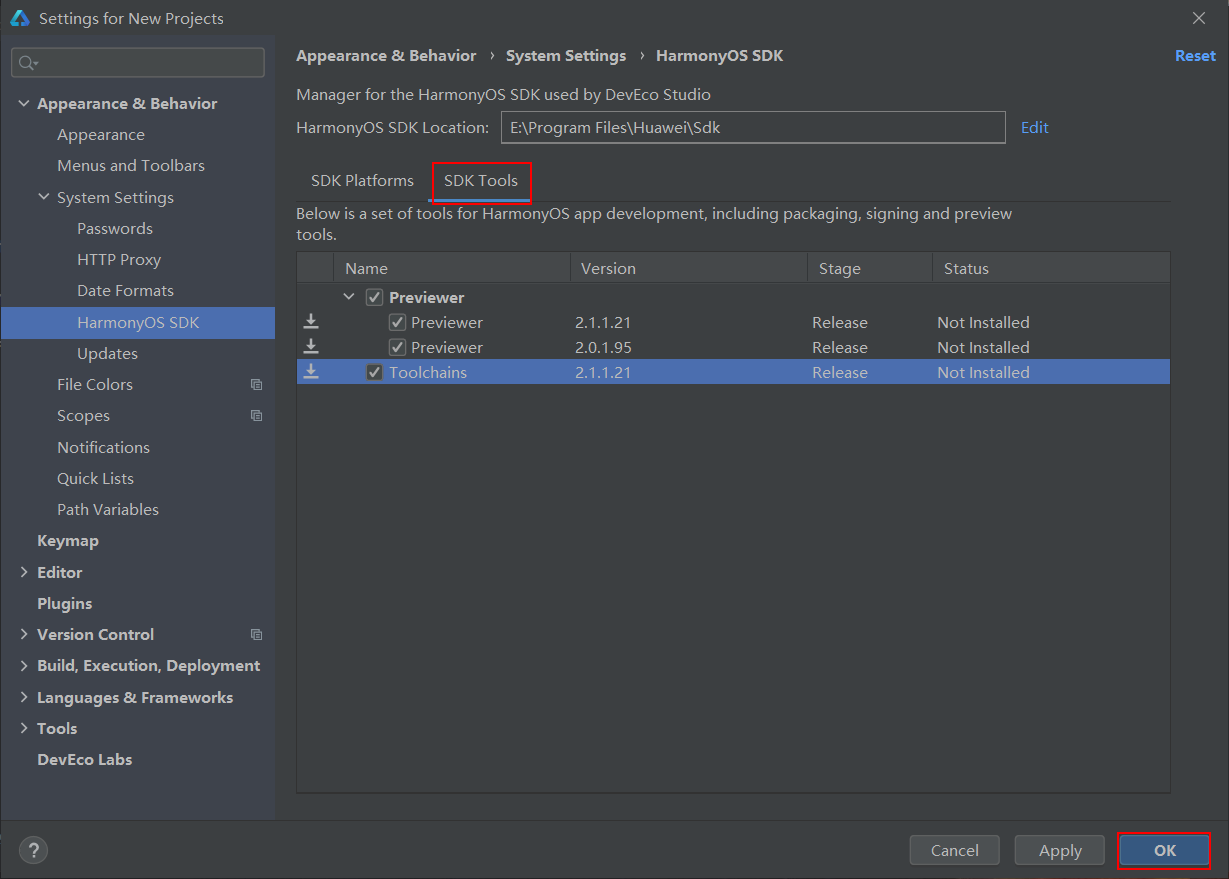


图7 配置安装SDK Tools中的Previewer组件和Toolchains组件

在新打开的窗口中，确认要安装的组件，如图8所示。单击OK按钮进入到下一步。

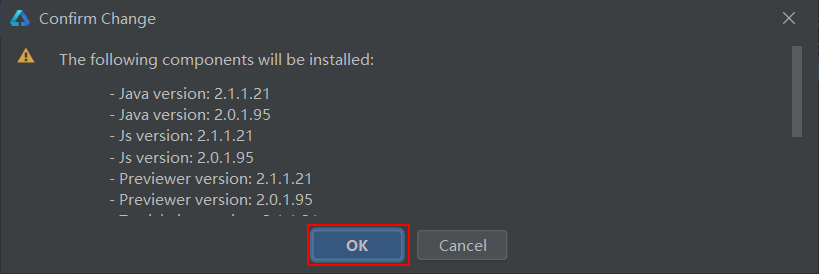


图8 确认要安装的组件

在新打开的窗口中，正在安装相关的组件，如图9所示。

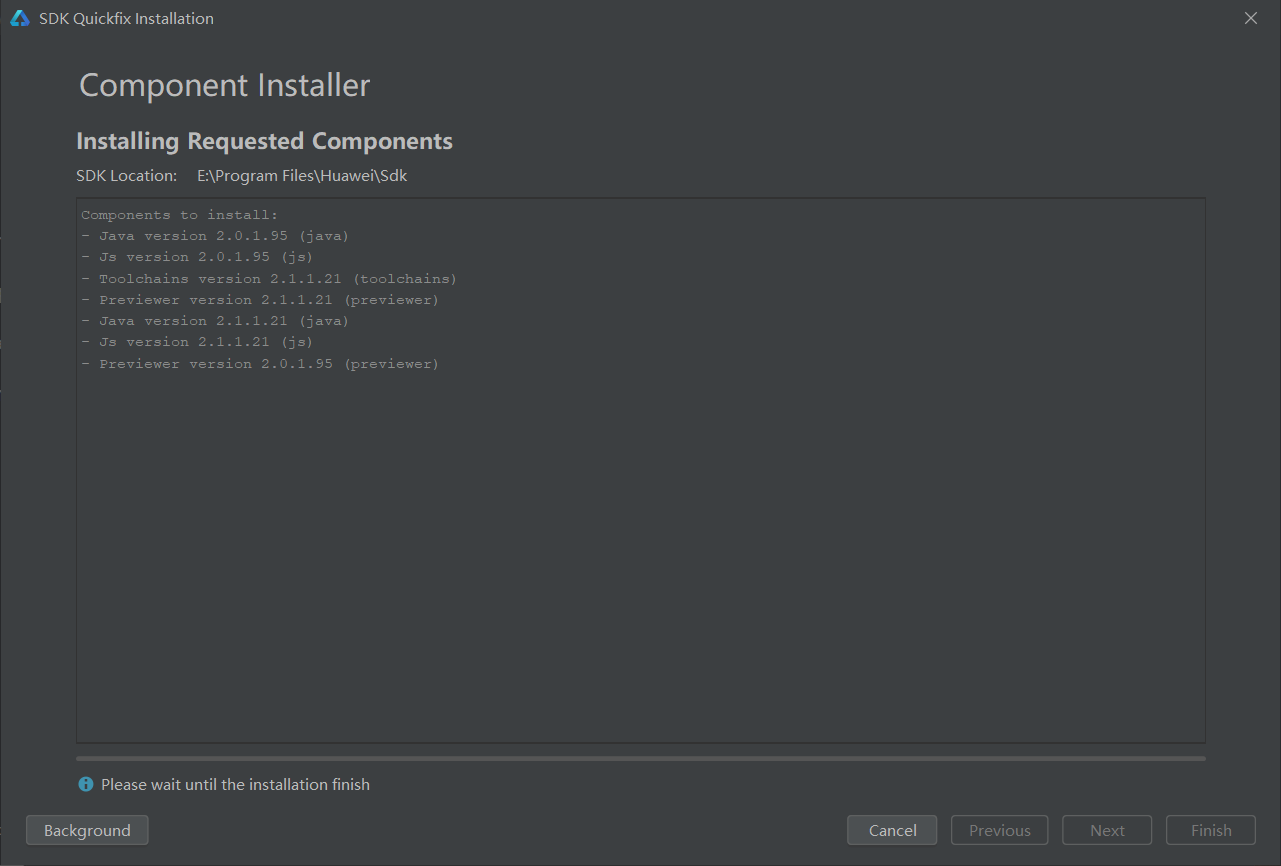


图9 正在安装相关的组件

安装完相关组件之后，打开一个新窗口，如图10所示。单击Finish按钮进入到下一步。

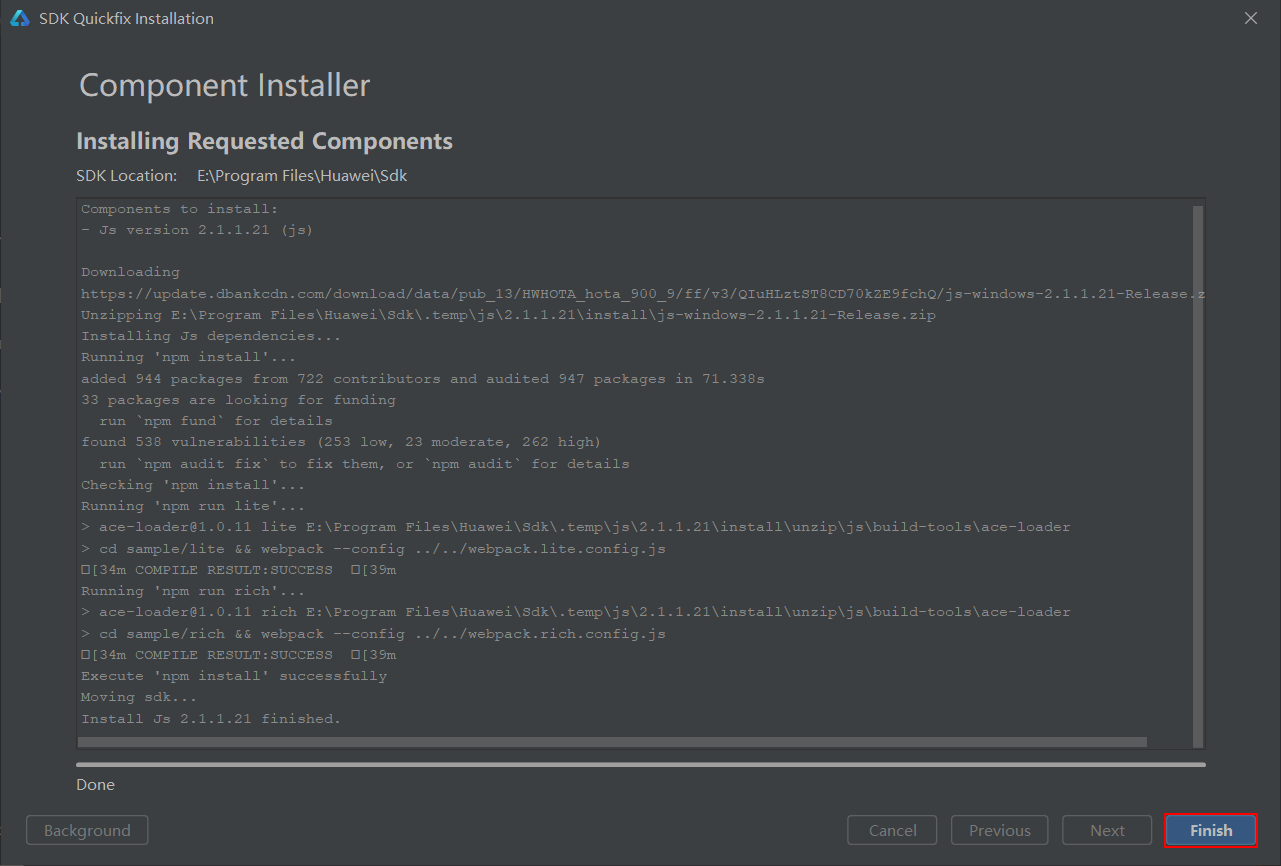


图10 相关组件安装完毕

在之前打开的窗口中，单击OK按钮，完成DevEco Studio的配置，如图11所示。

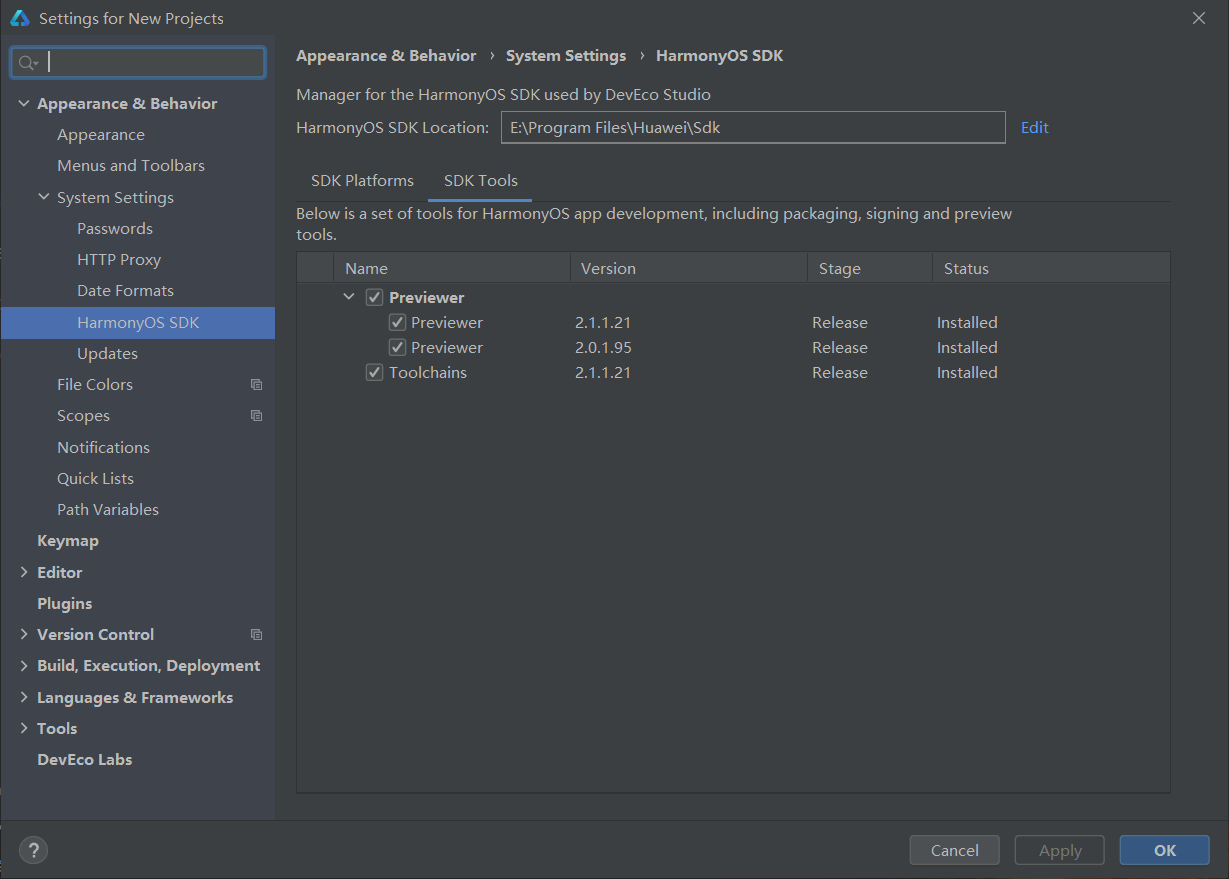


图11 完成DevEco Studio的配置

**2.2 Hello World**

搭建好开发环境之后,我们就可以新建一个 Hello World项目了。

打开集成开发环境DevEco Studio，单击Create HarmonyOS Project，以创建一个鸿蒙项目，如图12所示。

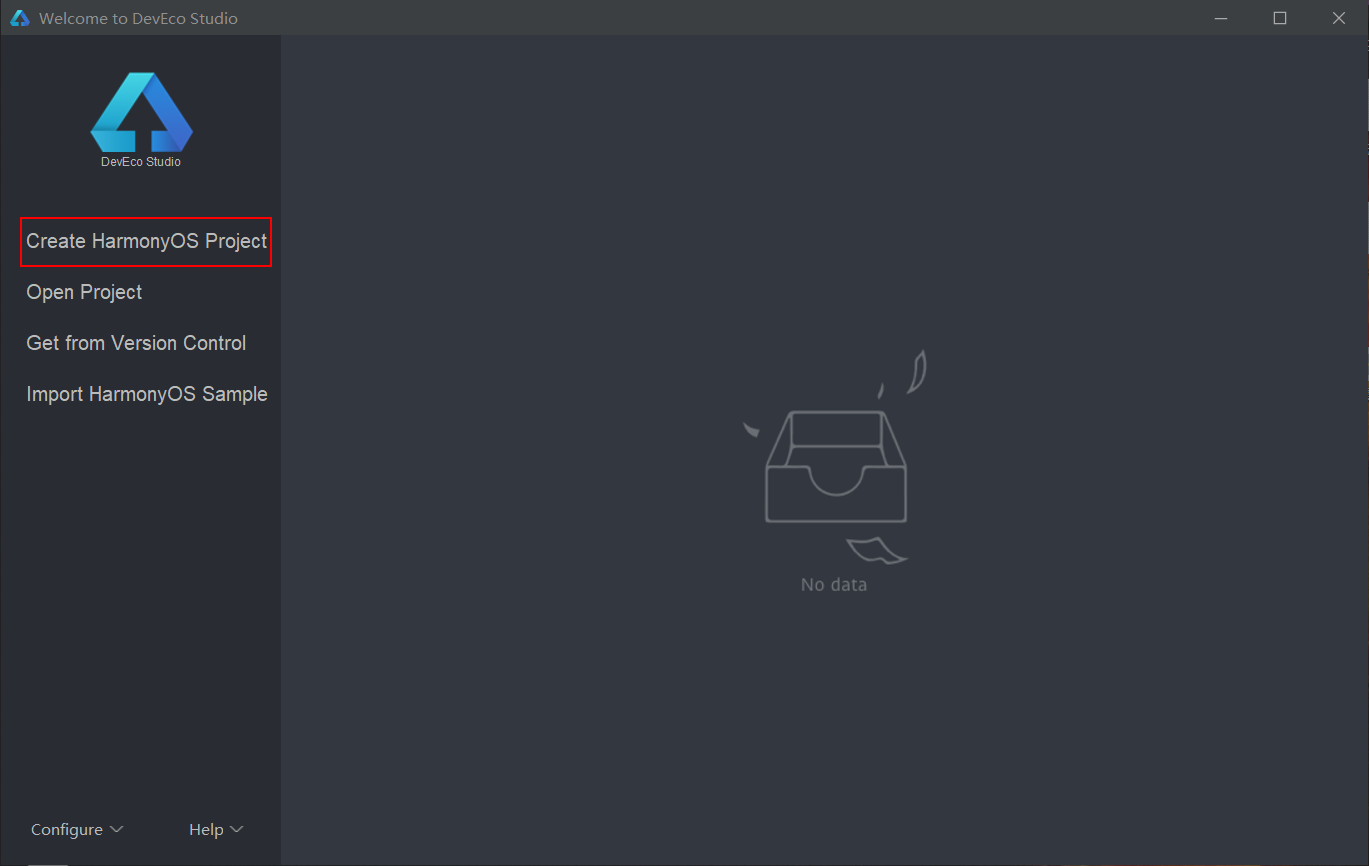


图12 创建一个鸿蒙项目

在新打开文件的窗口中，可以看到DevEco Studio已经为开发者提供了很多模板，如图13所示。首先名字中有Empty的模板仅仅为开发者提供了显示Hello World的应用程序；其它的模板为开发者提供了一些带有逻辑处理的应用程序。其次以“（JS）”结尾的模板既可以可以使用编程语言JavaScript，也可以使用编程语言Java；以“（Java）”结尾的模板只能使用编程语言Java。

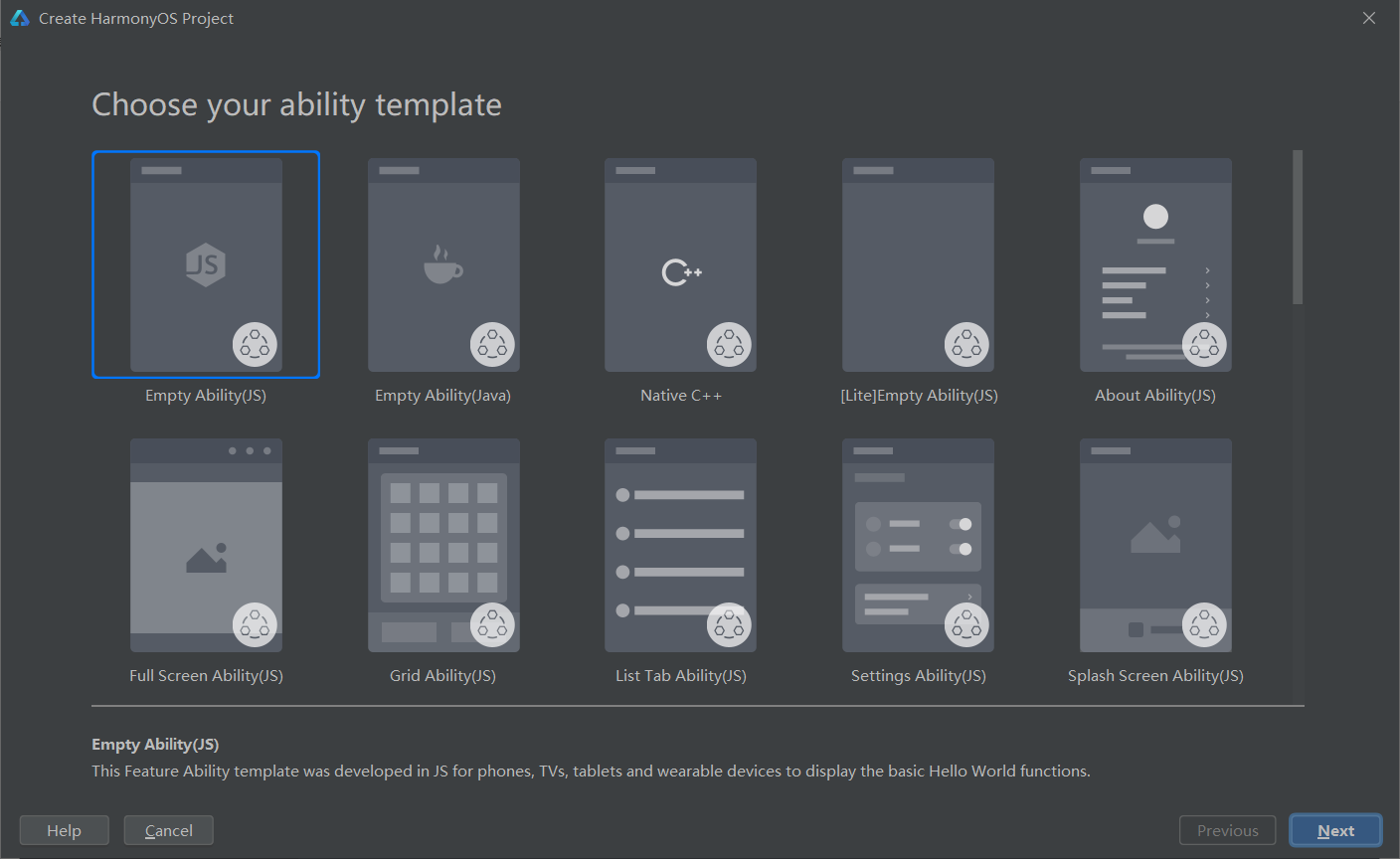


图13 DevEco Studio提供的模板

在此窗口中，将鼠标放在一个模板上，模板可以显示支持设备的图标，将鼠标放在Empty Ability(JS)上，可以看到此模板支持手机（Phone）、平板（Tablet）、电视（TV）和手表（Wearable），如图14所示。

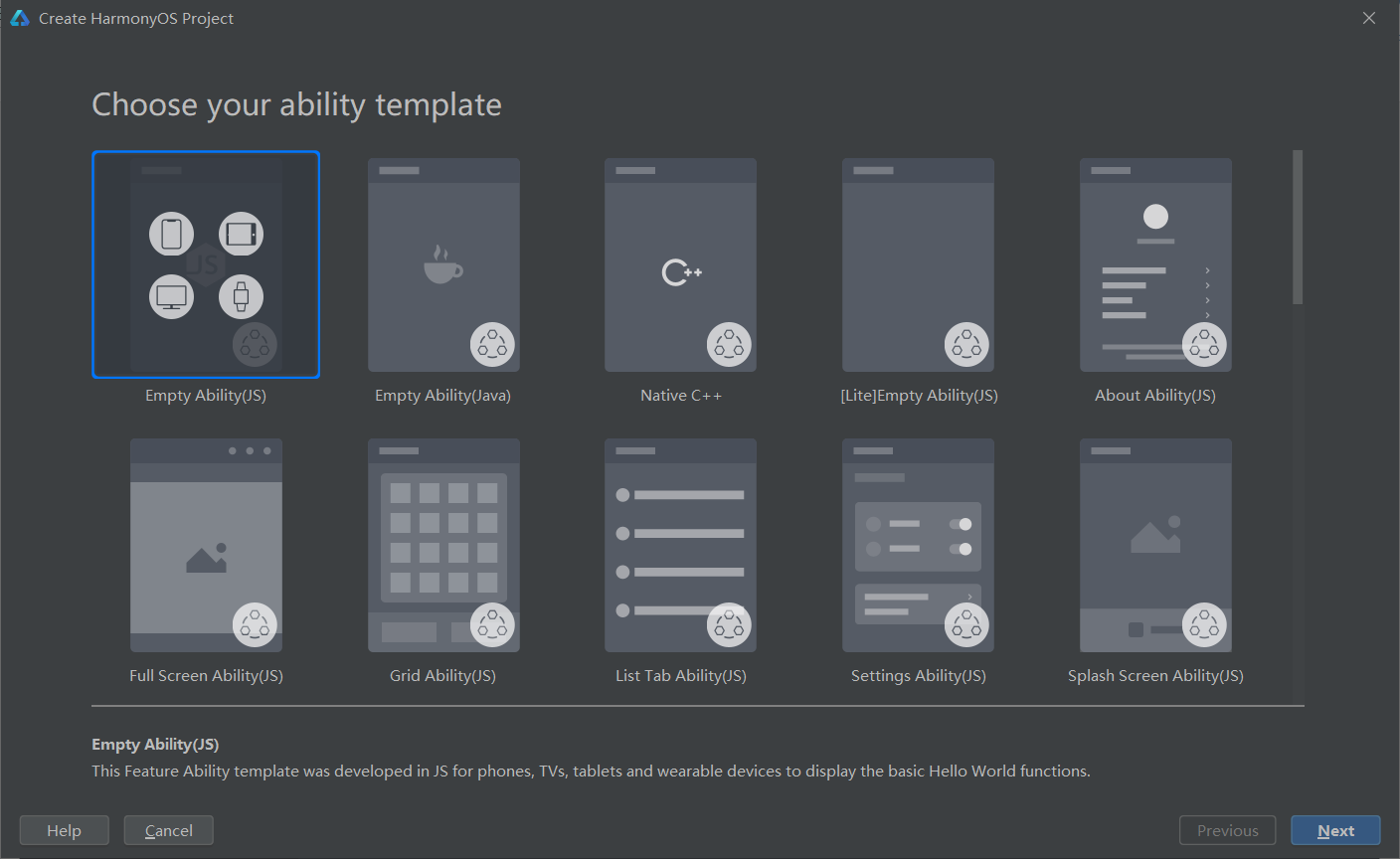


图14 Empty Ability(JS)模板支持的设备

将鼠标放在Empty Ability(Java)上，可以看到此模板支持手机（Phone）、平板（Tablet）、电视（TV）、手表（Wearable）和车机（Car），如图15所示。

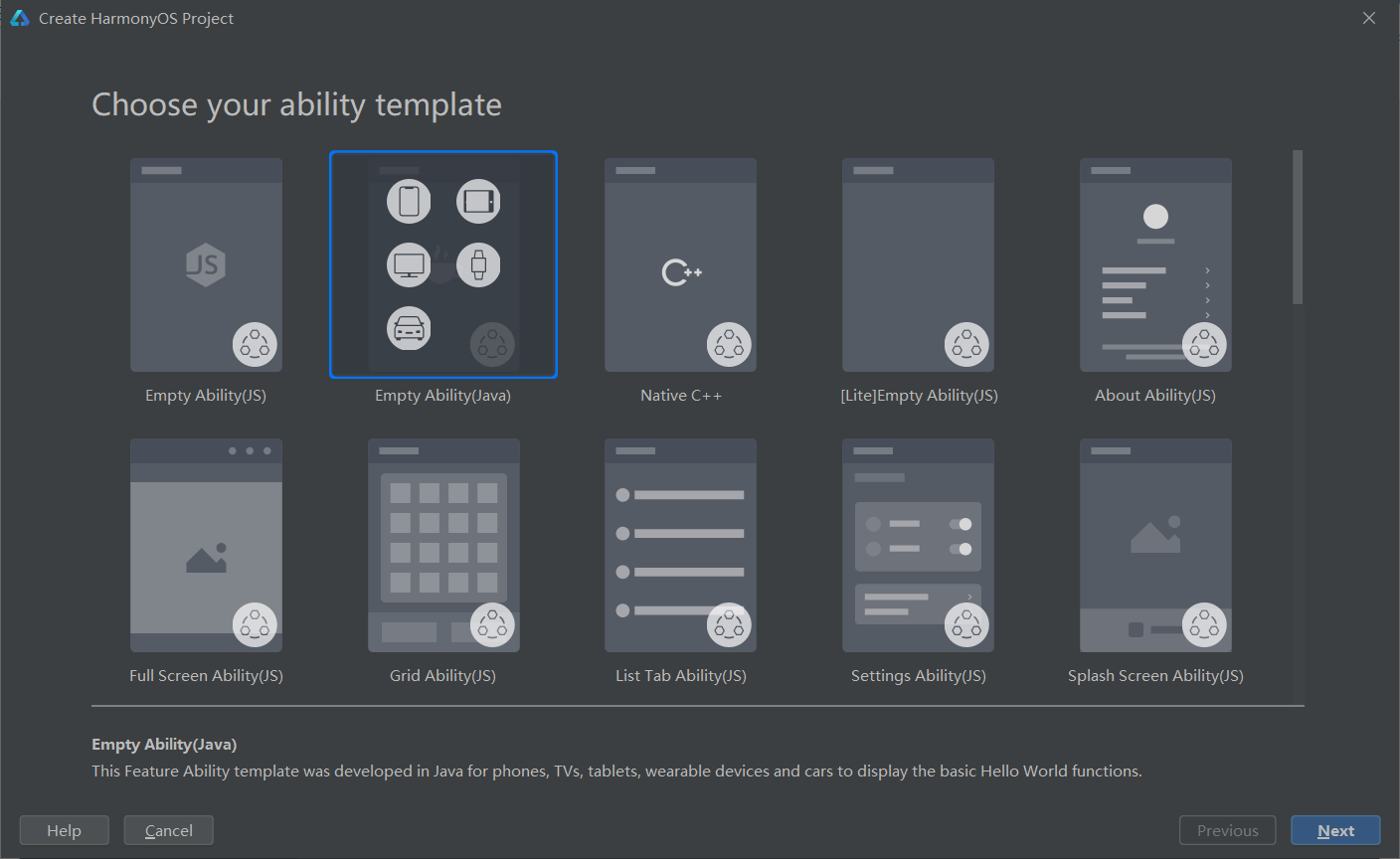


图15 Empty Ability(Java)模板支持的设备

将鼠标放在Native C++上，可以看到此模板支持手机（Phone）和车机（Car），如图16所示。

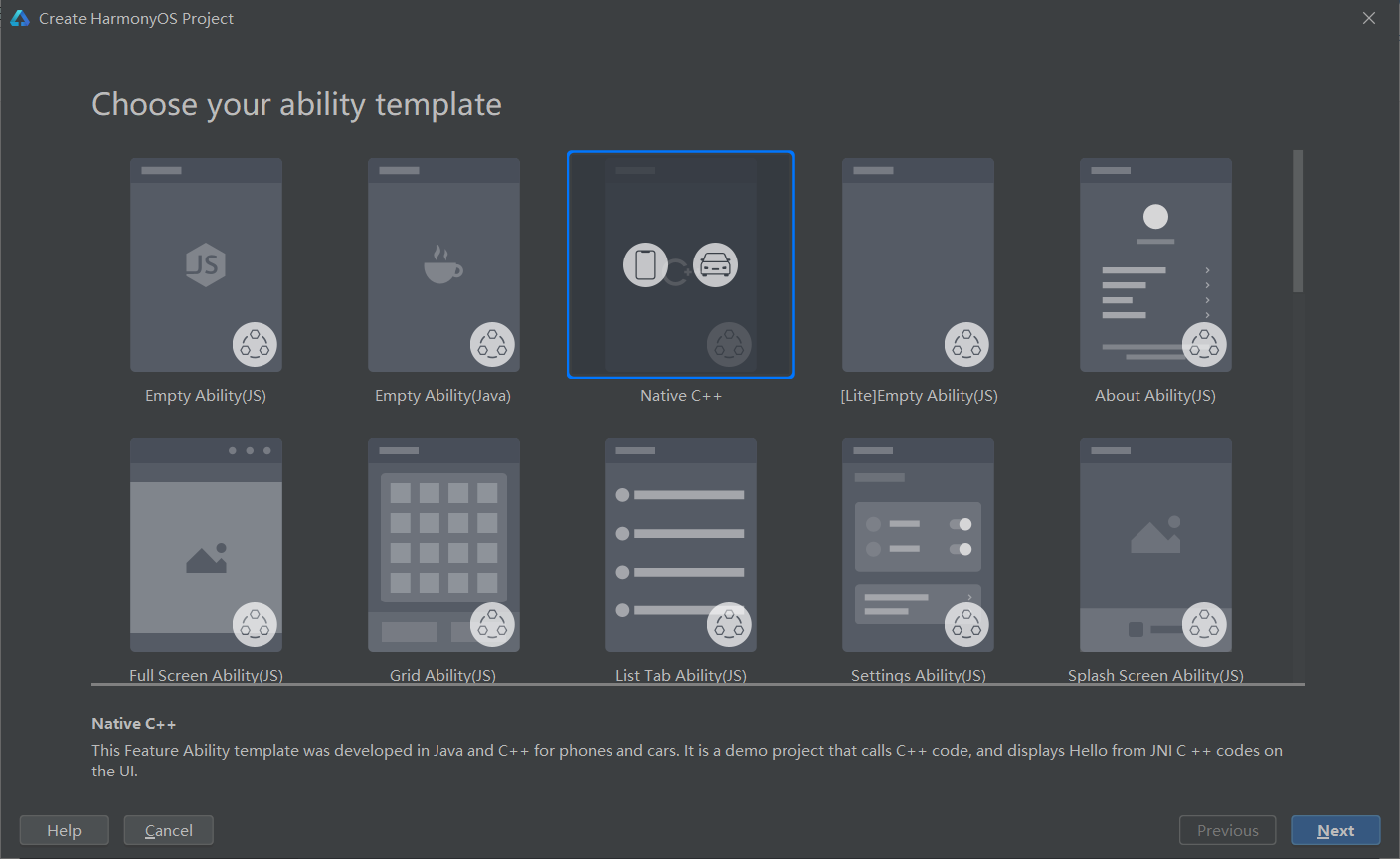


图16 Native C++模板支持的设备

将鼠标放在[Lite]Empty Ability(JS)上，可以看到此模板支持运动手表（Lite Wearable）和照相机（Smart Vision），如图17所示。

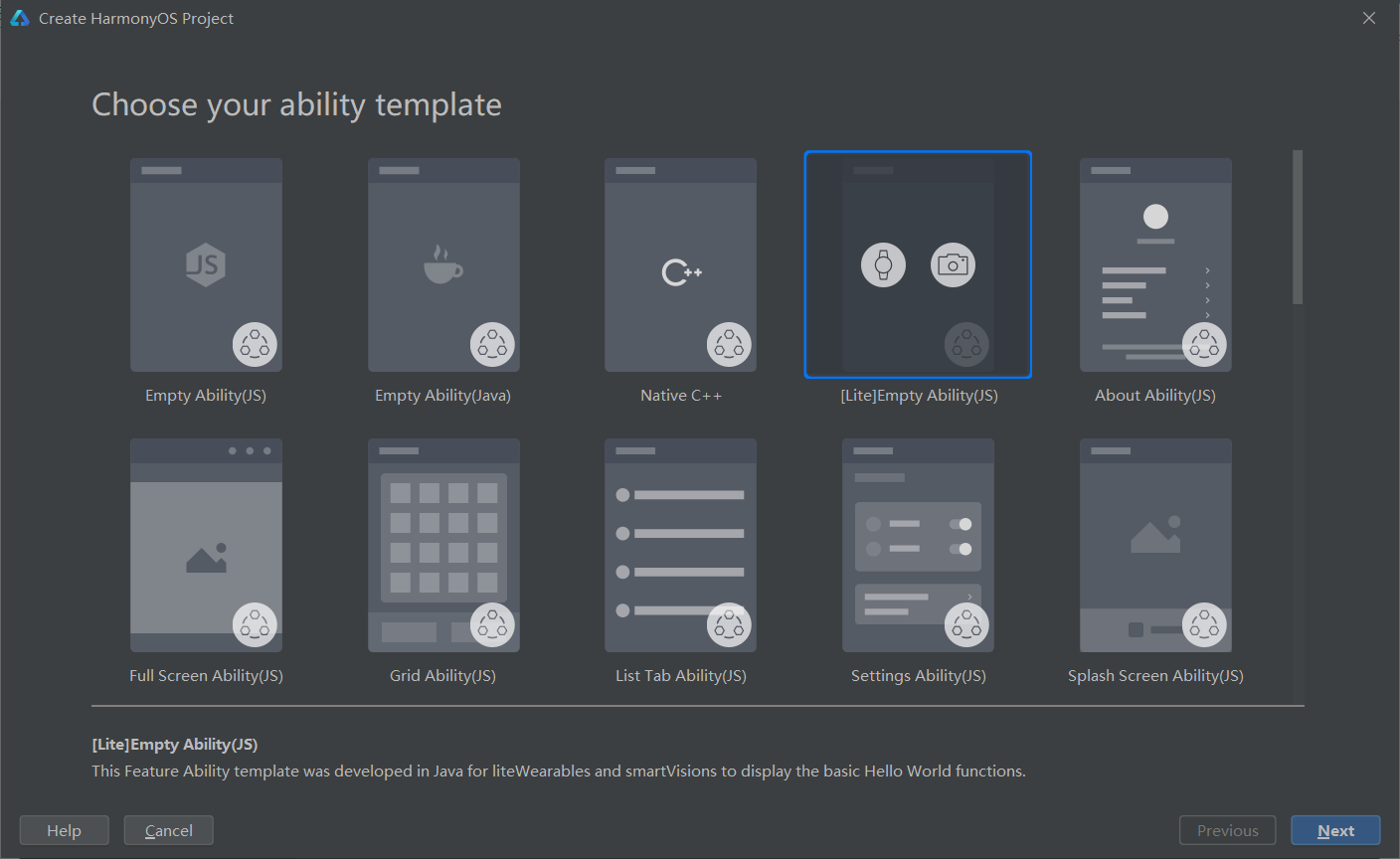


图17 [Lite]Empty Ability(JS)模板支持的设备

可见想要开发Lite Wearable上的App我们要选择[Lite]Empty Ability(JS)模板，并且只能使用编程语言JavaScript，编程语言Java不支持在Lite Wearable上的开发。

综上所述，华为鸿蒙已经可以在手机（Phone）、平板（Tablet）、电视（TV）、手表（Wearable）、车机（Car）、运动手表（Lite Wearable）和照相机（Smart Vision）上开发应用程序了。

如果我们开发的是TV或 Wearable 上的App，由于目前（截至2020年12月1日）华为还没有开放基于X86的本机模拟器，因此需要将编写的代码发送到远程的 ARM处理器运行代码。在本机上智能预览运行结果，而无法运行和调试代码。

如果我们开发的是 Lite Wearable上的 App，那么既可以使用本机的预览器Previewer 来预览代码的运行效果，也可以使用本机的模拟器Simulator来运行和调试代码，这给开发人员带俩了相当出色的体验！此外，当读者看到本书的时候，华为的智能手表没准已经上市了。在Lite Wearable平台上，相关的设备和开发工具是最成熟最完善的，因此，本书详细讲解的项目是运行在Lite Wearable上的。本书会跟随华为鸿蒙产品和开发工具包的发布节奏，在后续的版本中不断更新和扩充相应的实战项目。

本书的项目选择的是[Lite]Empty Ability(JS)模板，选择[Lite]Empty Ability(JS)的模板卡片，点击Next按钮进入下一步，如图18所示。

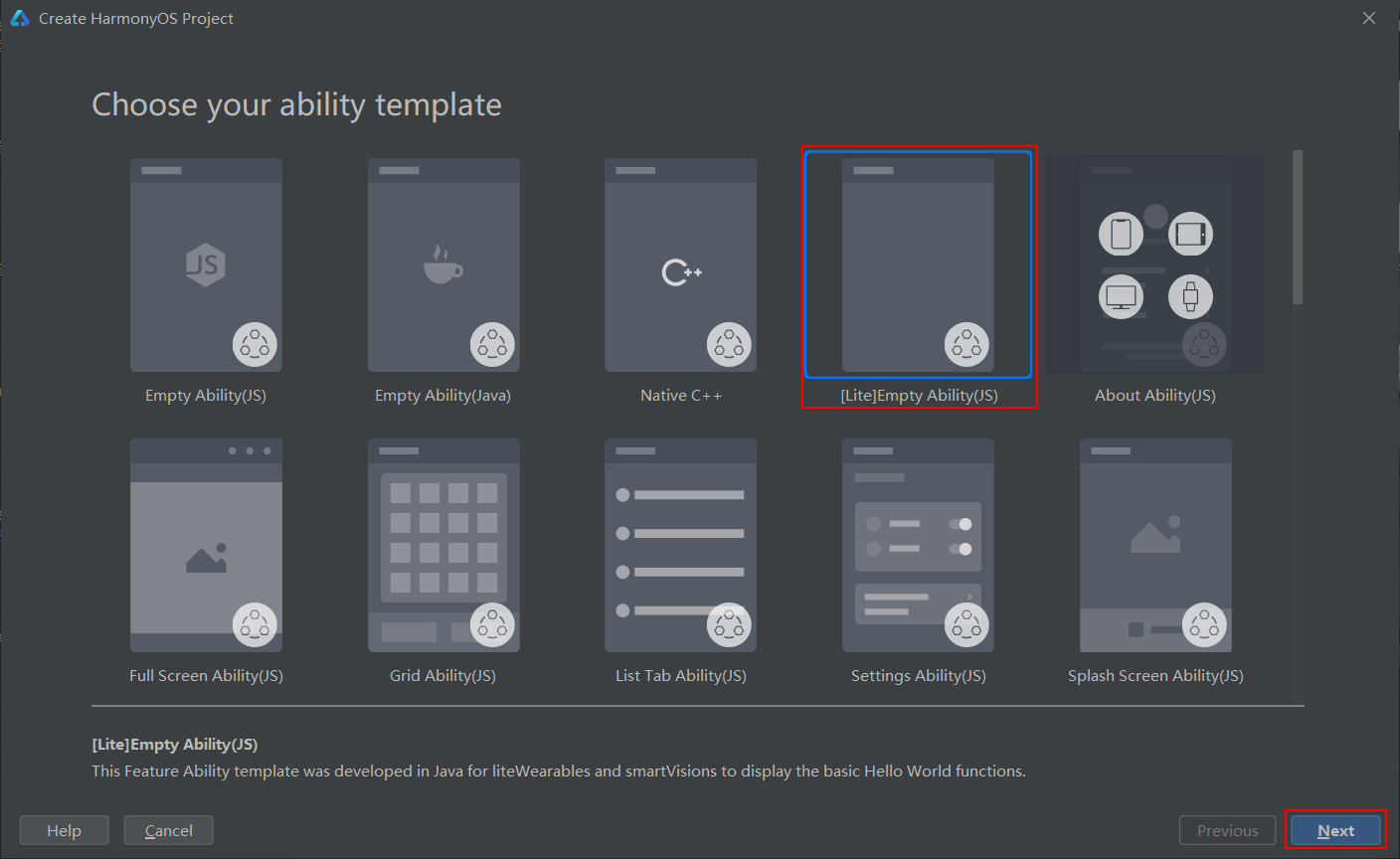


图18 选择[Lite]Empty Ability(JS)模板

在新打开的窗口中配置新建的项目，需要分别配置项目名（Project Name）、项目类型（Project Type）、包名（Package Name）、项目的保存位置（Save Location）、可兼容的SDK（Compatible API Version）和设备类型（Device Type）。将Project Name取名为BreathTraining，Project Type选择Application复选框，DevEco Studio会自动帮我们生成一个Package Name，其名称为com.example.breathtraining，Save Location（项目保存位置）读者根据自己的情况更改即可，Compatible API Version使用默认配置，Device Type选择Lite Wearable复选框，如图19所示。

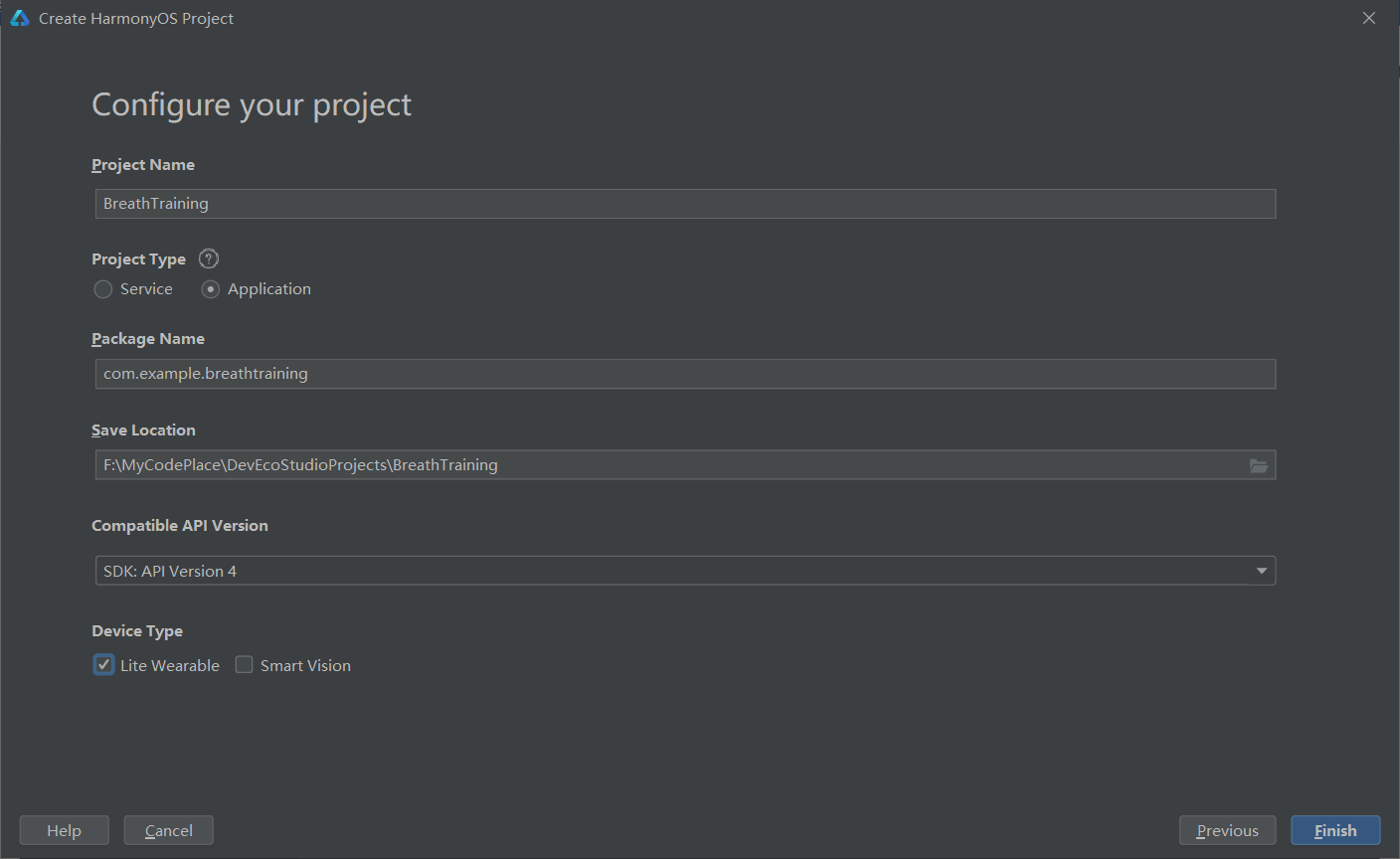


图19 配置新键的项目

单击Finish按钮之后，就创建了轻量级可穿戴设备的Hello World项目，也就是创建了一个运行在运动手表上的Hello World项目，如图20所示。

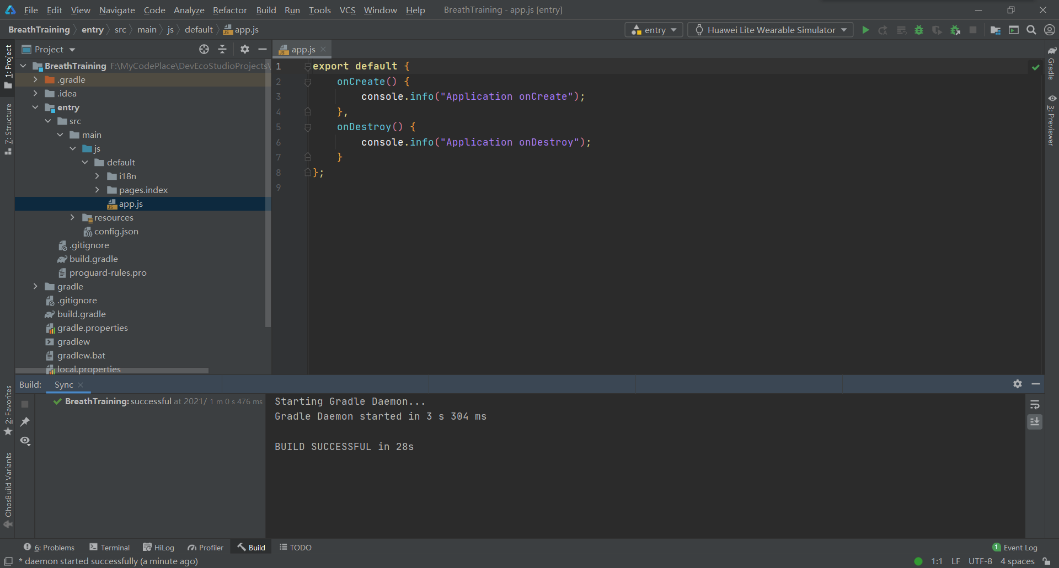


图20 新键的Hello World项目

选中菜单栏中的 View,在展开的菜单中选中Tool Windows，在展开的子菜单中单击Previewer命令，如图21所示。

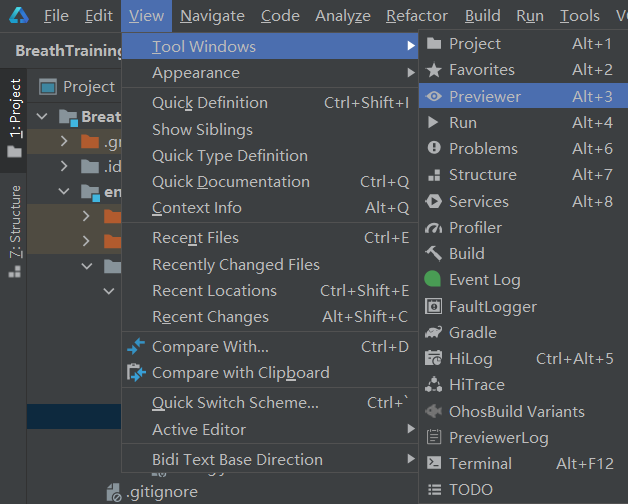


图21 菜单栏中的菜单项Previewer

通过Previewer就可以预览App的运行效果了。在智能手表的主页面显示了文本“Hello World”，如图22所示。

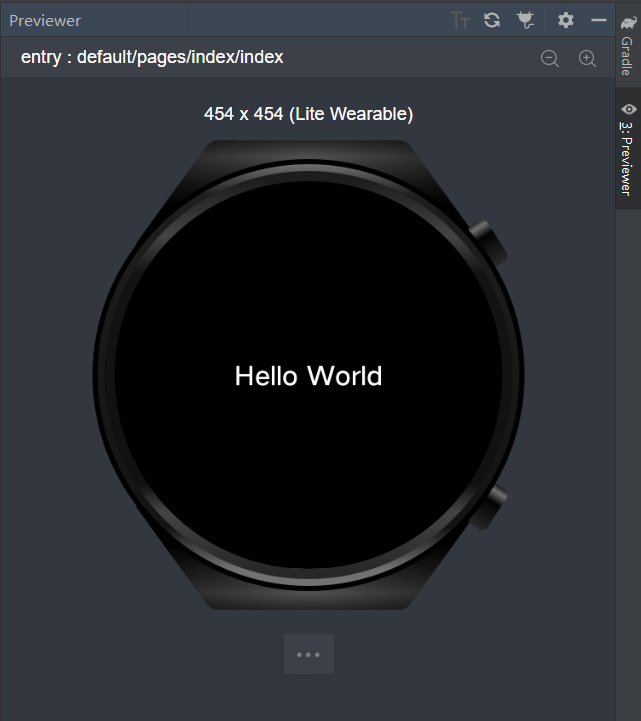


图22 Previewer

为了更方便地操作Previewer，可以单击右上角的Show Options Menu设置按钮，如图23所示。

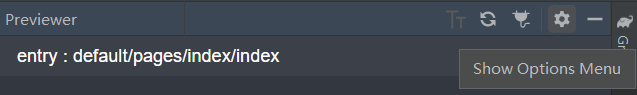


图23 Previewer

在打开的选项菜单中选中View Mode,然后单击Window命令，如图24所示。

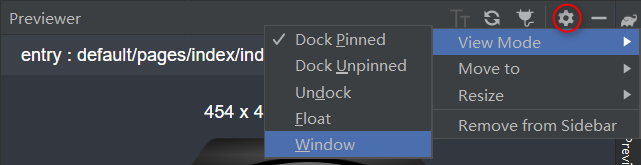


图24 设置Previewer的显示模式

这样，Previewer就会显示为一个非模态的浮动窗口——既可以将其最小化到任务栏，也可以将其最大化，如图25所示。

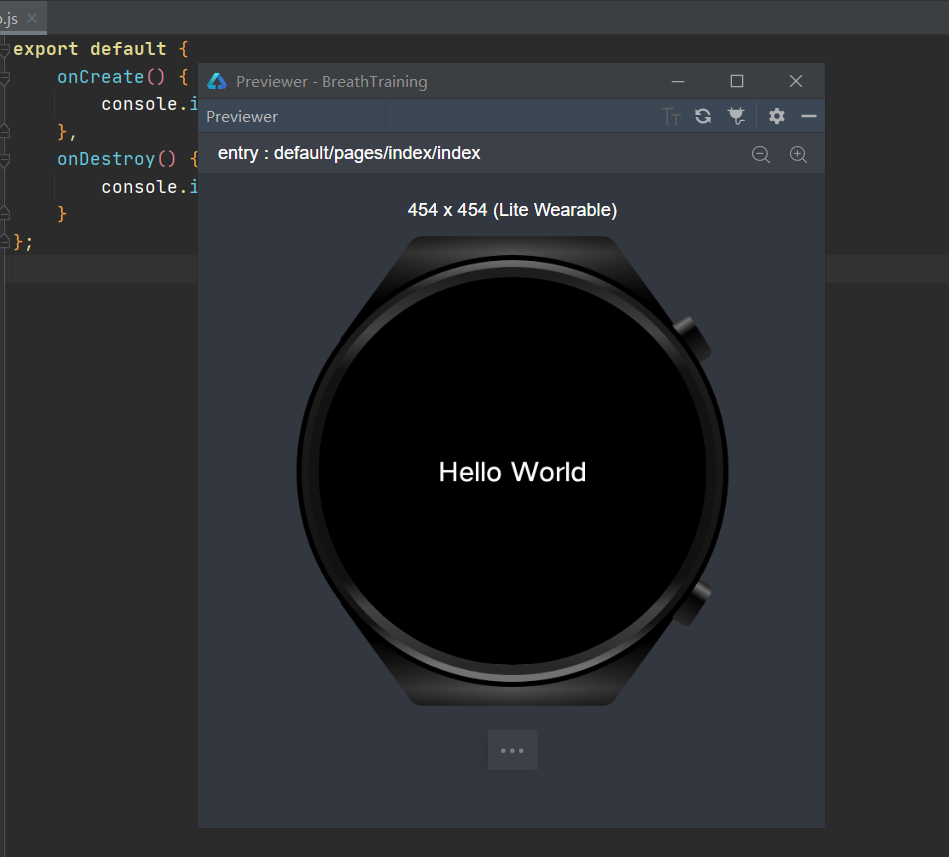


图25 显示为非模态浮动窗口的Previewer

单击手表下方的省略按钮，会在窗口的右侧列出Previewer的众多设置选项，例如屏幕亮度、心率、步数、地理位置、音量等，如图26所示。

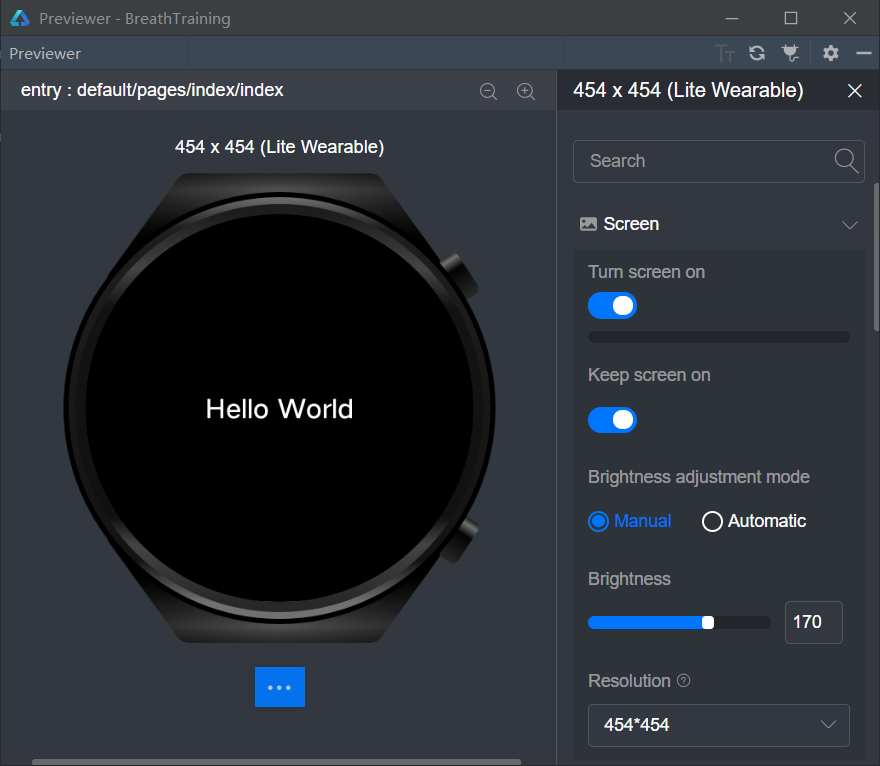


图26 Previewer的众多设置选项

在此点击省略按钮，Previewer的设置选项就被隐藏起来了。

2.2 Hello World （第27页 第四段）

**存在问题**

现在Javascript已经支持了ES6。

**修订内容**

轻量级智能穿戴支持的ES6语法有限，仅支持以下ES6 语法：

1. let/const
2. arrow functions
3. class
4. default value
5. destructuring assignment
6. destructuring binding pattern
7. enhanced object initializer
8. for-of
9. rest parameter
10. template strings

3.1 任务1：在主页面中添加一个按钮并响应其单击事件

**存在问题**

DevEco Studio的界面变化和Huawei Lite Wearable Simulato启动方式变化。

**修订内容**

在DevEco Studo的工具栏有一个下拉列表中选择Huawei Lite Wearable Simulator和一个小虫子图标样式的按钮，如图27和图28所示。



图27 Huawei Lite Wearable Simulator



图28 Debug应用的小虫子图标按钮

3.6 任务6：将主页面中选择器的值传递到训练页面

**存在问题**

用户如果不滑动选择器在训练页面显示不正常，如吸气/呼气不变化、坚持秒数显示错误和图标不旋转。

**修订内容**

为picker1value和picker2value的初值设置为“2”和“舒缓”，如代码清单1所示。

|  |
| --- |
| 代码清单1 index.js |
| import router from '@system.router';  **var picker1value = "2";**  **var picker2value = "舒缓";**  export default {  data: {  picker1range:["1", "2", "3"],  picker2range:["较慢", "舒缓", "较快"]  },  clickAction() {  router.replace({  uri:'pages/countdown/countdown',  params: {"data1": picker1value, "data2": picker2value}  });  },  changeAction1(pv) {  console.log("左边的选中中项：" + pv.newValue);  picker1value = pv.newValue;  },  changeAction2(pv) {  console.log("左边的选中中项：" + pv.newValue);  picker2value = pv.newValue;  },  ……  } |

3.23 任务23：在第2个训练报告页面中显示除心率曲线之外的所有内容

**存在问题**

最大心率的心率值显示不完整，如图29所示。



图29 心率最大值显示不全

**修订内容**

修改report2.css内容，在maxmin的类选择其中添加width样式，其值设置为50px。完整代码如代码清单2所示。

|  |
| --- |
| 代码清单2 report2.css |
| .container {  flex-direction: column;  justify-content: flex-start;  align-items: center;  width: 454px;  height: 454px;  }  .title-container {  justify-content: center;  align-items: center;  width: 300px;  height: 130px;  }  .title {  margin-top: 40px;  font-size: 38px;  }  .chart {  width: 400px;  height: 180px;  }  .list {  flex-direction: row;  width: 200px;  height: 45px;  }  .list-item {  justify-content: center;  align-items: center;  width: 100px;  height: 45px;  }  .icon {  width: 32px;  height: 32px;  }  .maxmin {  **width: 50px;**  font-size: 24px;  letter-spacing: 0px;  }  .average-container {  justify-content: space-between;  align-items: center;  width: 220px;  height: 55px;  }  .average {  font-size: 24px;  color: gray;  }  .average-number {  font-size: 38px;  letter-spacing: 0px;  } |

效果如图30所示。



图30 心率最大值显示完整

3.26 任务26：在第三个训练报告页面中显示除活动分布图之外的所有内容

**存在问题**

活动和静止百分比无法显示，如图31所示。



图31 活动和静止百分数没有显示

**修订内容**

修改report3.css中类选择器activity中justify-content的值，其值改为flex-start。Report3.css完整代码如代码清单3所示。

|  |
| --- |
| 代码清单3 report3.css |
| .container {  flex-direction: column;  justify-content: flex-start;  align-items: center;  width: 454px;  height: 454px;  }  .title-container {  justify-content: center;  align-items: center;  width: 300px;  height: 130px;  }  .title {  margin-top: 40px;  font-size: 38px;  }  .stack {  width: 340px;  height: 150px;  }  .chart-container {  width: 340px;  height: 150px;  }  .time-container {  width: 340px;  height: 25px;  justify-content: space-between;  align-items: center;  }  .time {  font-size: 18px;  color: gray;  letter-spacing: 0px;  }  .activities-list {  width: 180px;  height: 110px;  margin-top: 10px;  }  .activity{  width:175px;  height:45px;  **justify-content:flex-start;**  align-items:center;  }  .icon {  width: 32px;  height: 32px;  }  .type {  font-size: 24px;  letter-spacing: 0px;  margin-right: 10px;  }  .percent {  font-size: 30px;  letter-spacing: 0px;  } |

效果如图32所示。



图32 活动和静止百分数显示完整

3.29 任务29：在第4个训练报告页面中显示除压力分布图之外的所有内容

**存在问题**

canvas组件在轻量级可穿戴设备中不可用，压力最大值显示不完整，如图33和图34所示。



图33 canvas在轻量级可穿戴设备中不再支持



图34 压力最大值显示不完整

**修订内容**

将report4.hml中的canvas改为div，class的值设置为chart。

在report4.css中将canvas类选择器重新命名为chart，并将width的值改为340px,height的值改为150px,删除类选择器list-item的justify-content和align-items两个样式。Report4.hml完整代码和report4.css完整代码如代码清单4和5所示。

|  |
| --- |
| 代码清单4 report4.hml |
| <div class="container" onswipe="toNextPage">  <div class="title-container">  <text class="title">  压力分布  </text>  </div>  ~~<canvas class=”canvas”>~~  ~~</canvas>~~  **<div class="chart"></div>**  <div class="time-container">  <text class="time" for="{{timeRange}}">  {{$item}}  </text>  </div>  <list class="list">  <list-item class="list-item" for="{{maxmin}}">  <image class="icon" src="/common/{{$item.iconName}}.png"/>  <text class="maxmin">  {{$item.mValue}}  </text>  </list-item>  </list>  </div> |

|  |
| --- |
| 代码清单5 report4.css |
| .container {  flex-direction: column;  justify-content: flex-start;  align-items: center;  width: 454px;  height: 454px;  }  .title-container {  justify-content: center;  align-items: center;  width: 300px;  height: 130px;  }  .title {  margin-top: 40px;  font-size: 38px;  }  .**chart** {  width: **340px;**  height: **150px;**  }  .time-container {  width: 383px;  height: 25px;  justify-content: space-between;  align-items: center;  }  .time {  font-size: 18px;  color: gray;  letter-spacing: 0px;  }  .list {  flex-direction: row;  width: 200px;  height: 45px;  margin-top: 30px;  }  .list-item {  **~~justify-content: center;~~**  **~~align-items: center;~~**  width: 100px;  height: 45px;  }  .icon {  width: 32px;  height: 32px;  }  .maxmin {  font-size: 30px;  letter-spacing: 0px;  } |

效果如图35所示。



图35 压力最大值显示完整

3.30 任务30：在压力分布页面显示绘制的压力分布图

**存在问题**

因为canvas组件不能使用，用chart组件绘制压力分布图，其中chart绘图和canvas有差异。

**修订内容**

修改report4.hml中“压力分布”的div组件的div组件为chart，class属性值设置为chart，options属性的值设置为“{{options}}”，datasets属性的值设置为“{{datasets}}”。report4.hml的完整代码如代码清单6所示。

|  |
| --- |
| 代码清单6 report4.hml |
| <div class="container" onswipe="toNextPage">  <div class="title-container">  <text class="title">  压力分布  </text>  </div>  **<chart class="chart" type="bar" options="{{options}}" datasets="{{datasets}}">**  **</chart>**  <div class="time-container">  <text class="time" for="{{timeRange}}">  {{$item}}  </text>  </div>  <list class="list">  <list-item class="list-item" for="{{maxmin}}">  <image class="icon" src="/common/{{$item.iconName}}.png"/>  <text class="maxmin">  {{$item.mValue}}  </text>  </list-item>  </list>  </div> |

修改report4.js文件。

删除生命周期函数onShow，因为函数体是渲染canvas组件的。

增加chart的options和datasets数据。

在data中将options占位符的值初始化为一个字典，该字典中包含两个元素，分别设置x轴和y轴得参数。第一个元素得key事故xAxis，对应得value是一个字典。该字典中只包含一个元素，对应得key和value分别是axisTick和1，用于设置x轴上得刻度数量。在options对应得字典中，第二个元素得key是yAxis，对应得value是一个由两个元素组成得字典，分别用于设置y轴得最大值和刻度数量。其中，两个元素得key分别是max和axisTick，对应的value分别是100和1。

在data中将 datasets占位符的值初始化为一个字典的数组,该数组中只包含一个字典。该字典中只包含一个元素，元素的key是 data，对应的 value是一个数组[]，其中有48个元素进行占位，用于指定今日活动分布图中的数据。

在onInit函数中增加三行代码，根据pressures中产生的随机数据，为datasets赋值。

report4.js的完整代码如代码清单7所示。

|  |
| --- |
| 代码清单7 report4.js |
| import router from '@system.router';  var pressures = [];  export default {  data: {  timeRange: ["00:00", "06:00", "12:00", "18:00", "24:00"],  maxmin: [  {  iconName: "",  mValue: 0  },  {  iconName: "",  mValue: 0  },  ],  **options: {**  **xAxis: {**  **axisTick: 1**  **},**  **yAxis: {**  **max: 100,**  **axisTock: 1,**  **}**  **},**  **datasets: [**  **//总共有48组数据，每组为半个小时**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  **{**  **fillColor: '#888888',**  **data: [100],**  **},**  ],  },  onInit() {  for (let i = 0; i < 48; i++) {  pressures.push(this.getRandomInt(1, 99));  }  this.maxmin[0].mValue = Math.max.apply(null, pressures);  this.maxmin[1].mValue = Math.min.apply(null, pressures);  this.maxmin[0].iconName = "max-" + this.getColorNameByValue(this.maxmin[0].mValue);  this.maxmin[1].iconName = "min-" + this.getColorNameByValue(this.maxmin[1].mValue);  **for (let i = 0; i < 48; i++) {**  **let value = pressures[i];**  **this.datasets[i].fillColor = this.getColorHexByValue(value);**  **this.datasets[i].data = [value];**  **}**  },  getRandomInt(min, max) {  return Math.floor(Math.random() \* (max - min + 1) ) + min;  },  getColorNameByValue(value) {  if (value >= 80 && value <= 99) {  return "orange";  } else if (value >= 60 && value <= 79) {  return "yellow";  } else if (value >= 30 && value <= 59) {  return "cyan";  } else if (value >= 1 && value <= 29) {  return "blue";  }  },  getColorHexByValue(value) {  if (value >= 80 && value <= 99) {  return "#ffa500";  } else if (value >= 60 && value <= 79) {  return "#ffff00";  } else if (value >= 30 && value <= 59) {  return "#00ffff";  } else if (value >= 1 && value <= 29) {  return "#0000ff";  }  },  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/report5/report5'  });  break;  case 'bottom':  router.replace({  uri: 'pages/report3/report3'  });  }  }  } |

效果如图36所示



图36 压力分布图正确显示

3.34 任务34：在最大摄氧量界面的对应弧形和角度上显示星号

**存在问题**

canvas组件不可用。

**修订内容**

使用div+text，配合margin定位。div+text组件，叠加在stack最上面，div背景透明，用来定位text，text使用的字号为38px。

定位根据最大摄氧量数值计算margin-top和margin-left，基础是因为margin-top: 80px和margin-top:80是一样的效果，也就是说像素中的px字样可以省略。

style中的数据绑定，只能使用data中的数据，因为这些是实时更新的。

原来的定位坐标xy，其中的x就是margin-left，其中的y就是margin-top，现在需要把两个变量分开即可。

在report5.hml中，删除原来的canvas，改为div+text。

report5.hml的完整代码如代码清单8所示。

|  |
| --- |
| 代码清单8 report5.hml |
| <stack class="stack" onswipe="toNextPage">  <div class="container">  <div class="title-container">  <text class="title">  最大摄氧量  </text>  </div>  <div class="number-container">  <image if="{{isShow}}" class="number-icon" src="/common/{{first}}.png"/>  <image class="number-icon" src="/common/{{second}}.png"/>  </div>  <text class="unit">  ml/kg/min  </text>  <text class="level">  {{level}}水平  </text>  </div>  <progress class="progress" type="arc" percent="100" for="{{styles}}" style="color:{{$item.color}};start-angle:{{$item.startAngle}};"/>  **<div class="pointer">**  **<text class="pointer-txt" style="margin-top: {{y}};margin-left: {{x}};">**  **\***  **</text>**  **</div>**  </stack> |

在report5.css中增加div和text的相关类分别为pointer和pointer-txt类选择器。

删除canvas类选择器。

添加一个名为pointer的类选择器，以设置report5.hml中div组件的样式。将width和height的值都设置为454px，将background-color的值设置为transparent。

添加一个名为pointer-txt的类选择器，以设置report5.hml中text组件的样式。将font-size的值设置为38px，将color的值设置为yellow。

report5.css的完整代码如代码清单9所示。

|  |
| --- |
| 代码清单9 report5.css |
| .stack {  width: 454px;  height: 454px;  }  .container {  flex-direction: column;  justify-content: flex-start;  align-items: center;  width: 454px;  height: 454px;  }  .title-container {  justify-content: center;  align-items: center;  width: 300px;  height: 130px;  }  .title {  margin-top: 40px;  font-size: 38px;  }  .number-container {  width: 160px;  height: 180px;  justify-content: center;  align-items: center;  }  .number-icon {  width: 80px;  height: 100px;  margin-top: 40px;  }  .unit {  width: 200px;  height: 30px;  color: gray;  text-align: center;  font-size: 24px;  }  .level {  width: 200px;  height: 40px;  text-align: center;  margin-top: 30px;  }  .progress {  width: 454px;  height: 454px;  total-angle: 40deg;  stroke-width: 16px;  radius: 220px;  center-x: 224px;  center-y: 229px;  }  **.pointer{**  **width: 454px;**  **height: 454px;**  **background-color: transparent;**  **}**  **.pointer-txt{**  **font-size: 38px;**  **color: yellow;**  **}** |

在report5.js中，将x和y两个变量放到data中并赋值为“ ”。删除onShow函数，因为这是渲染canvas的函数。

report5.js完整的代码如代码清单10所示。

|  |
| --- |
| 代码清单10 report5.js |
| import router from '@system.router';  export default {  data: {  isShow: true,  first: "",  second: "",  level: "",  x: "",  y: "",  styles: [  {  color: "#ff0000",  startAngle: 220  },  {  color: "#ffa500",  startAngle: 260  },  {  color: "#ffd700",  startAngle: 300  },  {  color: "#ffff00",  startAngle: 340  },  {  color: "#adff2f",  startAngle: 20  },  {  color: "#00ffff",  startAngle: 60  },  {  color: "#4169e1",  startAngle: 100  },  ],  },  onInit() {  let vo2max = this.getRandomInt(1, 70);  let vo2max\_str = vo2max.toString();  if (vo2max\_str.length == 2) {  this.first = "num-" + vo2max\_str[0];  this.second = "num-" + vo2max\_str[1];  } else {  this.second = "num-" + vo2max\_str;  this.isShow = false;  }  this.level = this.getLevelByValue(vo2max);  let angle = -230 + vo2max \* (280 / 70);  this.x = Math.round(218 + 193 \* Math.cos(angle \* Math.PI / 180));  this.y = Math.round(218 + 193 \* Math.sin(angle \* Math.PI / 180));  },  getRandomInt(min, max) {  return Math.floor(Math.random() \* (max - min + 1) ) + min;  },  getLevelByValue(value) {  let levels = ["超低", "低", "较低", "一般", "高", "优秀", "卓越"];  let index = Math.floor((value - 1) / 10);  return levels[index];  },  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/contact/contact'  });  break;  case 'bottom':  router.replace({  uri: 'pages/report4/report4'  });  }  }  } |

效果如图37所示。



图37 最大摄氧量页面星号显示正常

**路由修复**

up换成了top，down换成了bottom。

路由存在问题的文件，修复代码如代码清单11所示。

|  |
| --- |
| 代码清单11 各个需要更改的路由 |
| //contact.js  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'bottom':  router.replace({  uri: 'pages/report5/report5'  });  }  }  //report1.js  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/report2/report2'  });  }  }  //report2.js  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/report3/report3'  })  case 'bottom':  router.replace({  uri: 'pages/report1/report1'  });  }  }  //report3.js  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/report3/report3'  })  case 'bottom':  router.replace({  uri: 'pages/report1/report1'  });  }  }  //report4.js  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/report5/report5'  });  break;  case 'bottom':  router.replace({  uri: 'pages/report3/report3'  });  }  }  //report5.js  toNextPage(e) {  switch(e.direction) {  case 'left':  router.replace({  uri: 'pages/index/index'  });  break;  case 'top':  router.replace({  uri: 'pages/contact/contact'  });  break;  case 'bottom':  router.replace({  uri: 'pages/report4/report4'  });  }  } |