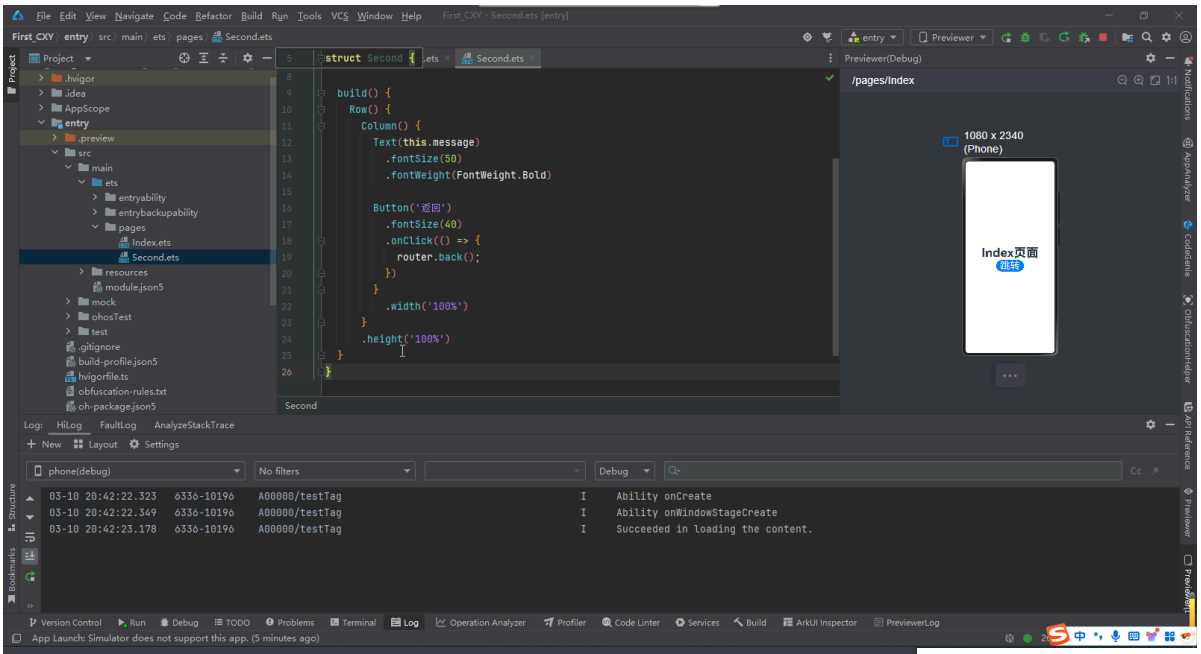
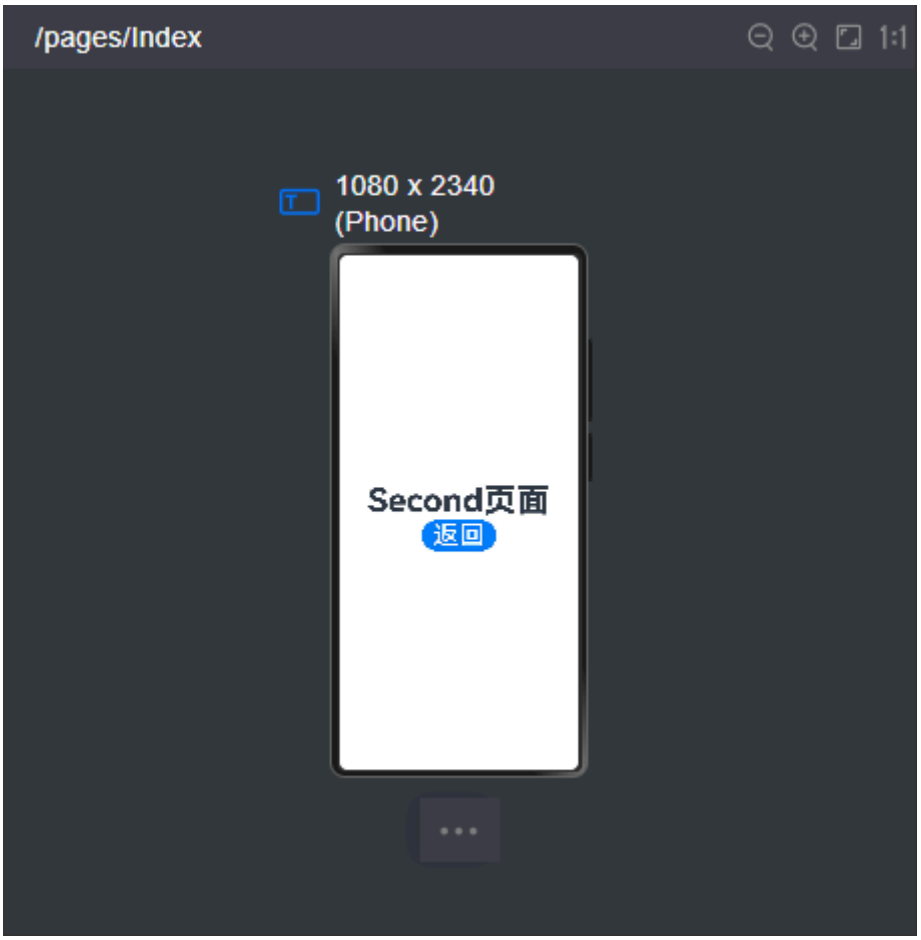
实验一：实现跳转+传输数据

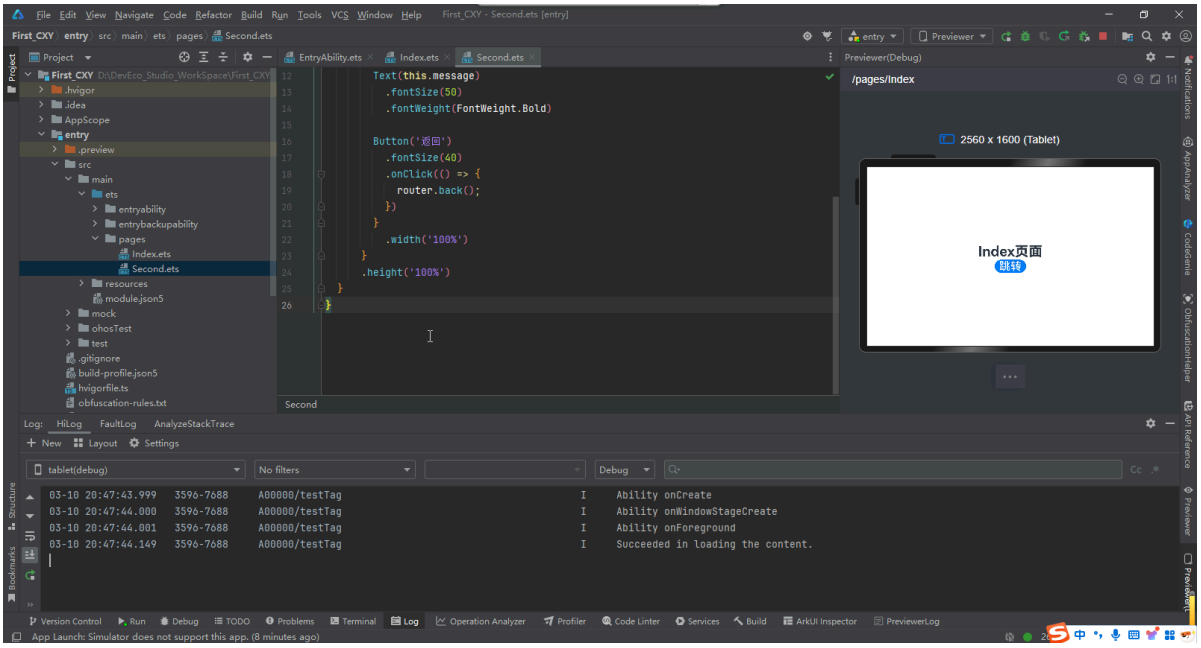
1、结合书上代码进行编写

以下是代码运行情况：




/pages/Index
1:1
1080 x 2340
(Phone)
Second页面
返回

改换别的设备运行


Text(this.message)
.fontSize(50)
.fontWeight(FontWeight.Bold)
Button('返回')
.fontSize(40)
.onClick(() => {
router.back();
})
.width('100%')
.height('100%')
/pages/Index
2560 x 1600 (Tablet)
Index页面
跳转
03-10 20:47:43.999 3596-7688 A00000/testTag I Ability onCreate
03-10 20:47:44.000 3596-7688 A00000/testTag I Ability onWindowStageCreate
03-10 20:47:44.001 3596-7688 A00000/testTag I Ability onForeground
03-10 20:47:44.149 3596-7688 A00000/testTag I Succeeded in loading the content.

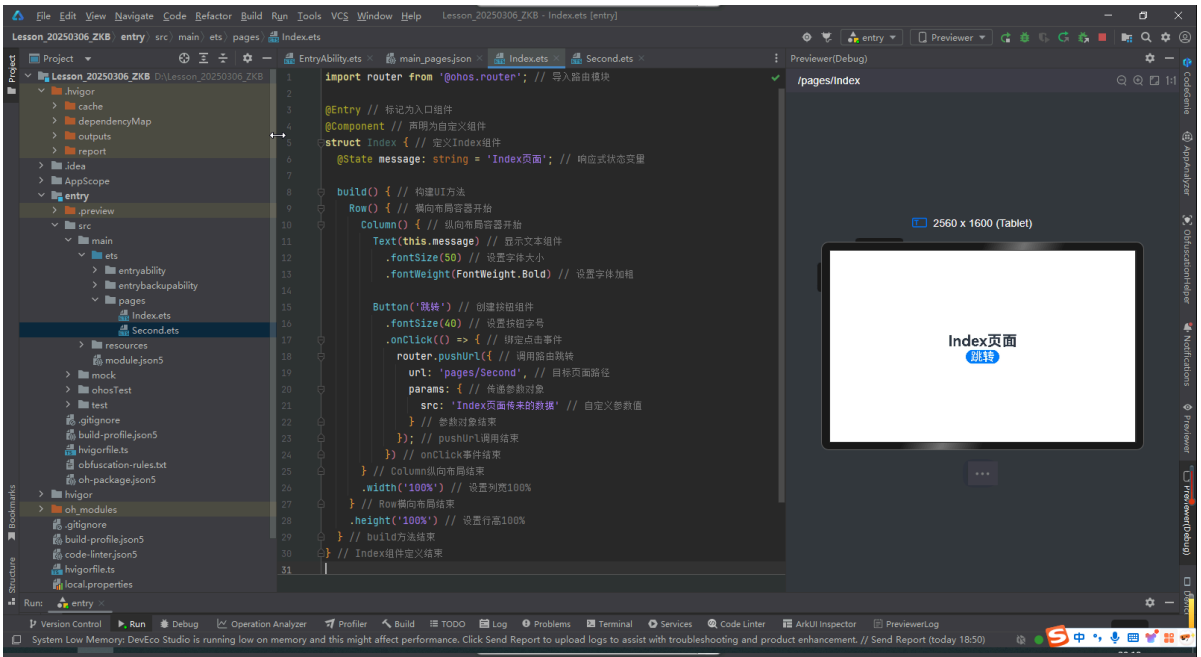
2、对上述代码逐行注释

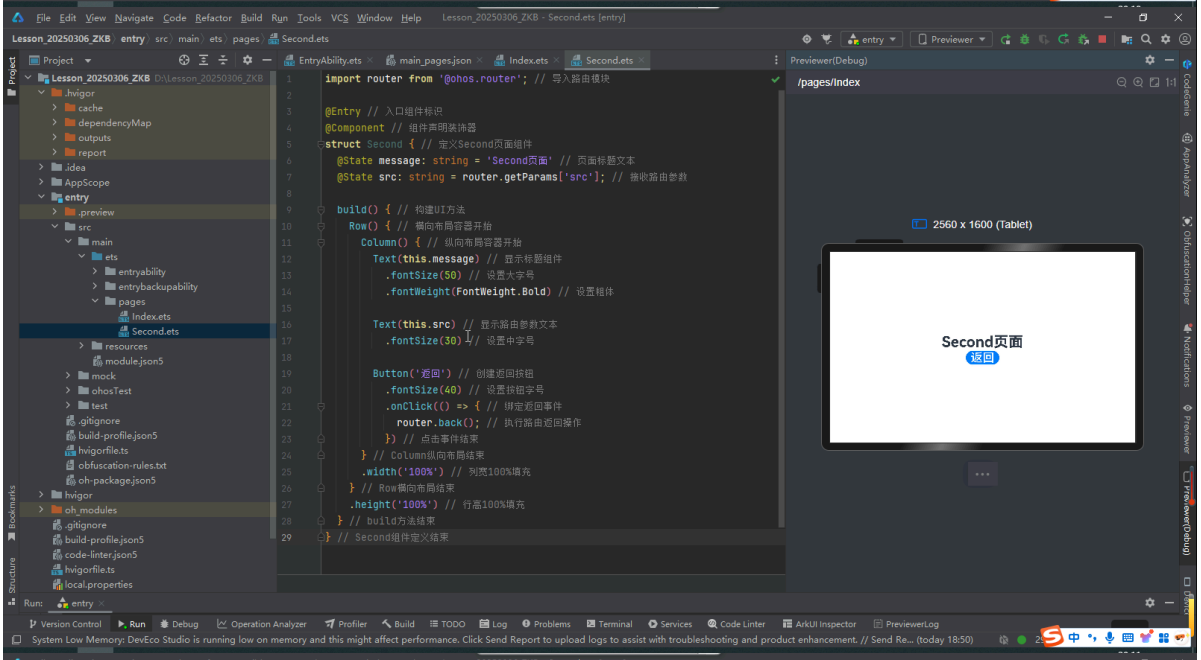
```
// Index页面
import router from '@ohos.router'; // 导入路由模块
```

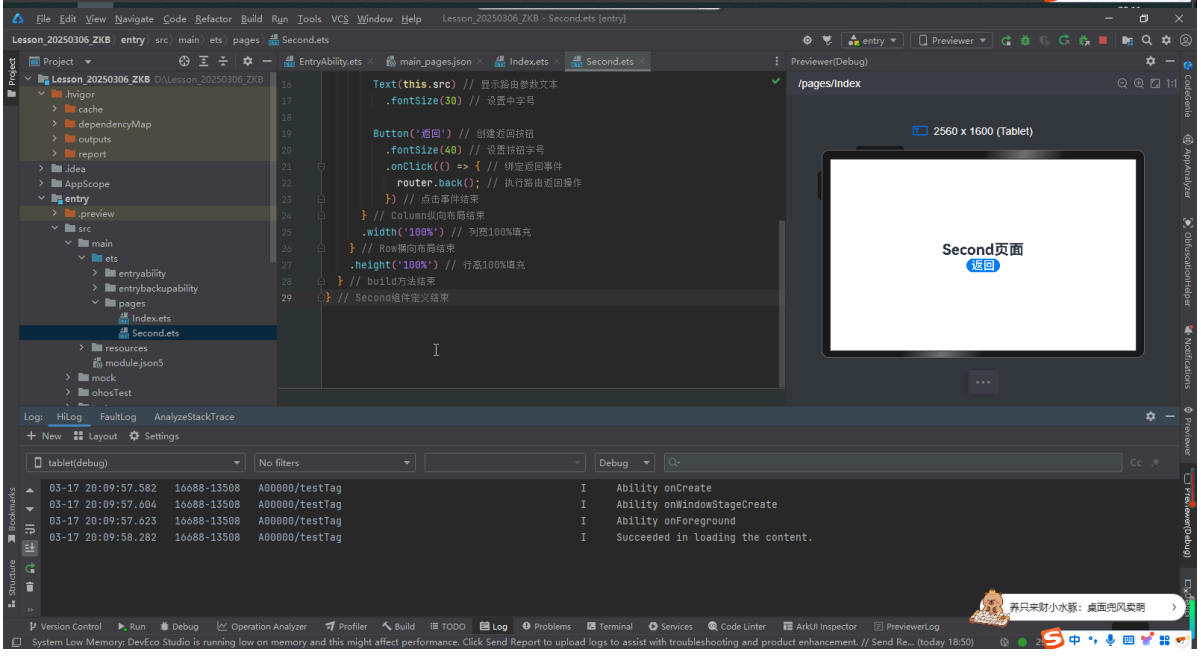
```
@Entry // 标记为入口组件
@Component // 声明为自定义组件
struct Index { // 定义Index组件
  @State message: string = 'Index页面'; // 响应式状态变量

  build() { // 构建UI方法
    Row() { // 横向布局容器开始
      Column() { // 纵向布局容器开始
        Text(this.message) // 显示文本组件
          .fontSize(50) // 设置字体大小
          .fontWeight(FontWeight.Bold) // 设置字体加粗

        Button('跳转') // 创建按钮组件
          .fontSize(40) // 设置按钮字号
          .onClick(() => { // 绑定点击事件
            router.pushUrl({ // 调用路由跳转
              url: 'pages/Second', // 目标页面路径
              params: { // 传递参数对象
                src: 'Index页面传来的数据' // 自定义参数值
              } // 参数对象结束
            }); // pushUrl调用结束
          }) // onClick事件结束
      } // Column纵向布局结束
      .width('100%') // 设置列宽100%
    } // Row横向布局结束
    .height('100%') // 设置行高100%
  } // build方法结束
} // Index组件定义结束
```

```
//Second页面
import router from '@ohos.router'; // 导入路由模块

@Entry // 入口组件标识
@Component // 组件声明装饰器
struct Second { // 定义Second页面组件
  @State message: string = 'Second页面' // 页面标题文本
  @State src: string = router.getParams['src']; // 接收路由参数

  build() { // 构建UI方法
    Row() { // 横向布局容器开始
      Column() { // 纵向布局容器开始
        Text(this.message) // 显示标题组件
          .fontSize(50) // 设置大字号
          .fontWeight(FontWeight.Bold) // 设置粗体

        Text(this.src) // 显示路由参数文本
          .fontSize(30) // 设置中字号

        Button('返回') // 创建返回按钮
          .fontSize(40) // 设置按钮字号
          .onClick(() => { // 绑定返回事件
            router.back(); // 执行路由返回操作
          }) // 点击事件结束
      } // Column纵向布局结束
```

```
      .width('100%') // 列宽100%填充
    } // Row横向布局结束
    .height('100%') // 行高100%填充
  } // build方法结束
} // Second组件定义结束
```


Index.ets
import router from '@ohos.router'; // 导入路由模块
@Entry // 标记为入口组件
@Component // 声明为自定义组件
struct Index { // 定义Index组件
@State message: string = 'Index页面'; // 响应式状态变量
build() { // 构建UI方法
Row() { // 横向布局容器开始
Column() { // 纵向布局容器开始
Text(this.message) // 显示文本组件
.fontSize(50) // 设置字体大小
.fontWeight(FontWeight.Bold) // 设置字体加粗
Button('跳转') // 创建按钮组件
.fontSize(40) // 设置按钮字号
.onClick(() => { // 绑定点击事件
router.pushUrl({ // 调用路由跳转
url: 'pages/Second', // 目标页面路径
params: { // 传递参数对象
src: 'Index页面传来的数据' // 自定义参数值
} // 参数对象结束
}); // pushUrl调用结束
}) // onClick事件结束
} // Column纵向布局结束
.width('100%') // 设置列宽100%
} // Row横向布局结束
.height('100%') // 设置行高100%
} // build方法结束
} // Index组件定义结束
Previewer(Debug)
/pages/Index
2560 x 1600 (Tablet)
Index页面
跳转


Second.ets
import router from '@ohos.router'; // 导入路由模块
@Entry // 入口组件标识
@Component // 组件声明装饰器
struct Second { // 定义Second页面组件
@State message: string = 'Second页面' // 页面标题文本
@State src: string = router.getParams()['src']; // 接收路由参数
build() { // 构建UI方法
Row() { // 横向布局容器开始
Column() { // 纵向布局容器开始
Text(this.message) // 显示标题组件
.fontSize(50) // 设置大字号
.fontWeight(FontWeight.Bold) // 设置粗体
Text(this.src) // 显示路由参数文本
.fontSize(30) // 设置中字号
Button('返回') // 创建返回按钮
.fontSize(40) // 设置按钮字号
.onClick(() => { // 绑定返回事件
router.back(); // 执行路由返回操作
}) // 点击事件结束
} // Column纵向布局结束
.width('100%') // 列宽100%填充
} // Row横向布局结束
.height('100%') // 行高100%填充
} // build方法结束
} // Second组件定义结束
Previewer(Debug)
/pages/Index
2560 x 1600 (Tablet)
Second页面
返回


Second.ets
Text(this.src) // 显示路由参数文本
.fontSize(30) // 设置中字号
Button('返回') // 创建返回按钮
.fontSize(40) // 设置按钮字号
.onClick(() => { // 绑定返回事件
router.back(); // 执行路由返回操作
}) // 点击事件结束
} // Column纵向布局结束
.width('100%') // 列宽100%填充
} // Row横向布局结束
.height('100%') // 行高100%填充
} // build方法结束
} // Second组件定义结束
Previewer(Debug)
/pages/Index
2560 x 1600 (Tablet)
Second页面
返回
03-17 20:09:57.582 16688-13508 A00000/testTag I Ability onCreate
03-17 20:09:57.604 16688-13508 A00000/testTag I Ability onWindowStageCreate
03-17 20:09:57.623 16688-13508 A00000/testTag I Ability onForeground
03-17 20:09:58.282 16688-13508 A00000/testTag I Succeeded in loading the content.

3、问题分析与修改

（1）发现问题

✅ 页面跳转成功
❌ 数据传输失败（目标页面无法接收参数）

（2）解决方案

1. 补充按钮点击事件绑定
2. 添加`router.pushUrl`参数传递
3. 验证目标页面参数接收

4、最终实现效果

（1）修改部分

①修正router.getParams的方法调用，添加括号。
②定义PageParams接口，明确参数结构。
③使用类型断言确保参数类型正确。
④通过 as PageParams 强制类型转换，将动态的 Object 转换为静态类型对象，提升类型安全性，防止运行时错误。

（2）最终结果

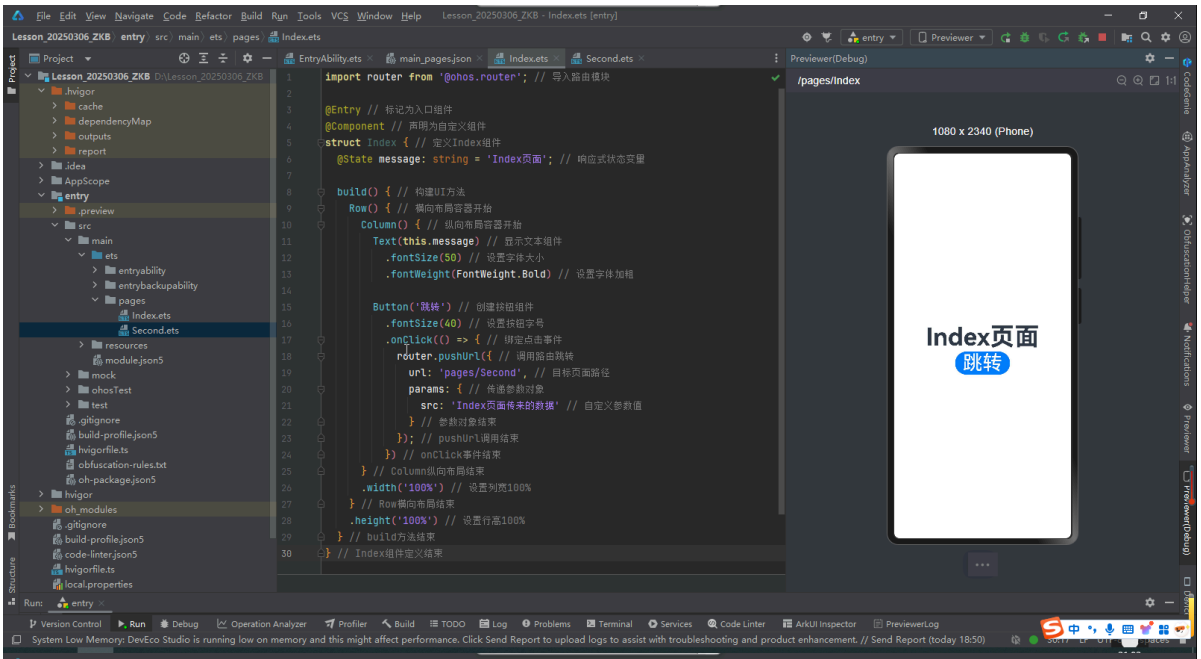
✅ 页面跳转成功
✅ 数据成功传输（目标可以接收）

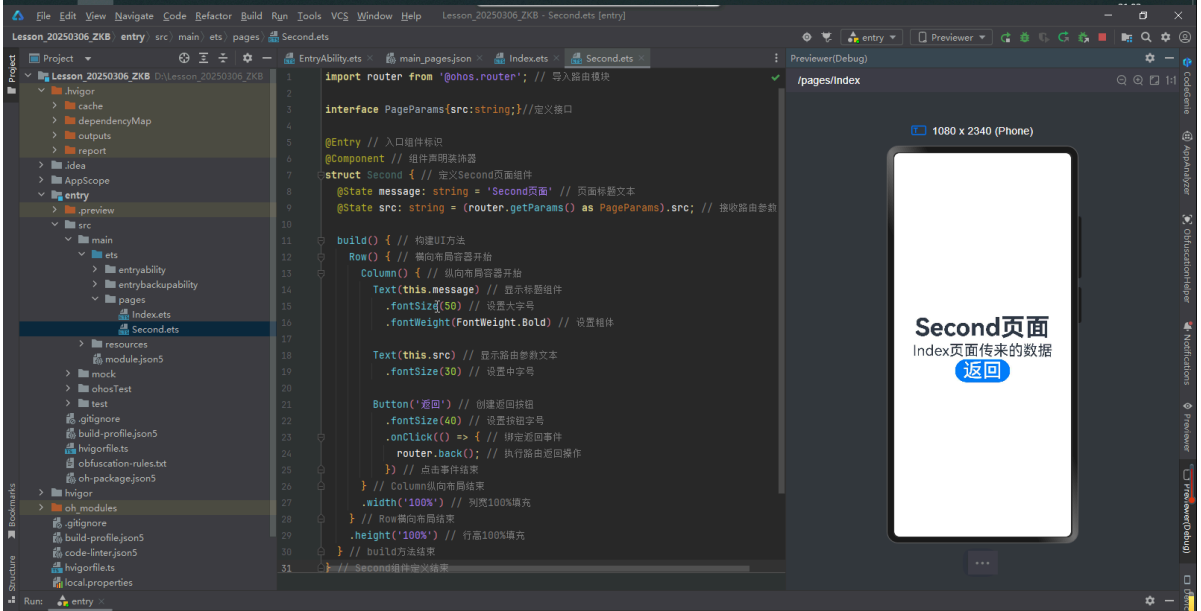
（3）完整版最终代码

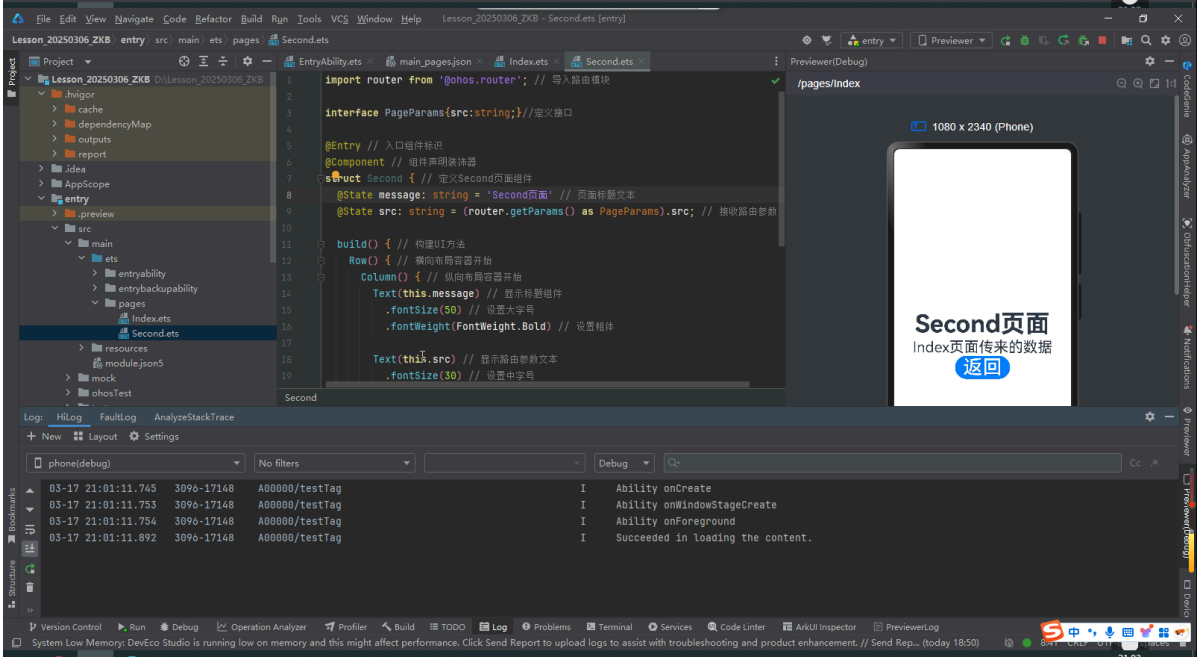
```
//Index页面
import router from '@ohos.router'; // 导入路由模块

@Entry // 标记为入口组件
@Component // 声明为自定义组件
struct Index { // 定义Index组件
  @State message: string = 'Index页面'; // 响应式状态变量

  build() { // 构建UI方法
    Row() { // 横向布局容器开始
      Column() { // 纵向布局容器开始
        Text(this.message) // 显示文本组件
          .fontSize(50) // 设置字体大小
          .fontWeight(FontWeight.Bold) // 设置字体加粗

        Button('跳转') // 创建按钮组件
          .fontSize(40) // 设置按钮字号
          .onClick(() => { // 绑定点击事件
            router.pushUrl({ // 调用路由跳转
              url: 'pages/Second', // 目标页面路径
              params: { // 传递参数对象
                src: 'Index页面传来的数据' // 自定义参数值
              } // 参数对象结束
```

```
            }); // pushUrl调用结束
          }) // onClick事件结束
      } // Column纵向布局结束
      .width('100%') // 设置列宽100%
    } // Row横向布局结束
    .height('100%') // 设置行高100%
  } // build方法结束
} // Index组件定义结束
```

```
//Second页面
import router from '@ohos.router'; // 导入路由模块

interface PageParams{src:string;}//定义接口

@Entry // 入口组件标识
@Component // 组件声明装饰器
struct Second { // 定义Second页面组件
  @State message: string = 'Second页面' // 页面标题文本
  @State src: string = (router.getParams() as PageParams).src; // 接收路由参数

  build() { // 构建UI方法
    Row() { // 横向布局容器开始
      Column() { // 纵向布局容器开始
        Text(this.message) // 显示标题组件
          .fontSize(50) // 设置大字号
          .fontWeight(FontWeight.Bold) // 设置粗体

        Text(this.src) // 显示路由参数文本
          .fontSize(30) // 设置中字号

        Button('返回') // 创建返回按钮
          .fontSize(40) // 设置按钮字号
          .onClick(() => { // 绑定返回事件
            router.back(); // 执行路由返回操作
          }) // 点击事件结束
      } // Column纵向布局结束
      .width('100%') // 列宽100%填充
    } // Row横向布局结束
    .height('100%') // 行高100%填充
  } // build方法结束
} // Second组件定义结束
```


import router from '@ohos.router'; // 导入路由模块
@Entry // 标记为入口组件
@Component // 声明为自定义组件
struct Index { // 定义Index组件
@State message: string = 'Index页面'; // 响应式状态变量
build() { // 构建UI方法
Row() { // 横向布局容器开始
Column() { // 纵向布局容器开始
Text(this.message) // 显示文本组件
.fontSize(50) // 设置字体大小
.fontWeight(FontWeight.Bold) // 设置字体加粗
Button('跳转') // 创建按钮组件
.fontSize(40) // 设置按钮字号
.onClick(() => { // 绑定点击事件
router.pushUrl({ // 调用路由跳转
url: 'pages/Second', // 目标页面路径
params: { // 传递参数对象
src: 'Index页面传来的数据' // 自定义参数值
} // 参数对象结束
}); // pushUrl调用结束
}) // onClick事件结束
} // Column纵向布局结束
.width('100%') // 设置列宽100%
} // Row横向布局结束
.height('100%') // 设置行高100%
} // build方法结束
} // Index组件定义结束
/pages/Index
1080 x 2340 (Phone)
Index页面
跳转


import router from '@ohos.router'; // 导入路由模块
interface PageParams{src:string;}//定义接口
@Entry // 入口组件标识
@Component // 组件声明装饰器
struct Second { // 定义Second页面组件
@State message: string = 'Second页面' // 页面标题文本
@State src: string = (router.getParams() as PageParams).src; // 接收路由参数
build() { // 构建UI方法
Row() { // 横向布局容器开始
Column() { // 纵向布局容器开始
Text(this.message) // 显示标题组件
.fontSize(50) // 设置大字号
.fontWeight(FontWeight.Bold) // 设置粗体
Text(this.src) // 显示路由参数文本
.fontSize(30) // 设置中字号
Button('返回') // 创建返回按钮
.fontSize(40) // 设置按钮字号
.onClick(() => { // 绑定返回事件
router.back(); // 执行路由返回操作
}) // 点击事件结束
} // Column纵向布局结束
.width('100%') // 列宽100%填充
} // Row横向布局结束
.height('100%') // 行高100%填充
} // build方法结束
} // Second组件定义结束
/pages/Index
1080 x 2340 (Phone)
Second页面
Index页面传来的数据
返回


import router from '@ohos.router'; // 导入路由模块
interface PageParams{src:string;}//定义接口
@Entry // 入口组件标识
@Component // 组件声明装饰器
struct Second { // 定义Second页面组件
@State message: string = 'Second页面' // 页面标题文本
@State src: string = (router.getParams() as PageParams).src; // 接收路由参数
build() { // 构建UI方法
Row() { // 横向布局容器开始
Column() { // 纵向布局容器开始
Text(this.message) // 显示标题组件
.fontSize(50) // 设置大字号
.fontWeight(FontWeight.Bold) // 设置粗体
Text(this.src) // 显示路由参数文本
.fontSize(30) // 设置中字号
03-17 21:01:11.745 3096-17148 A00000/testTag I Ability onCreate
03-17 21:01:11.753 3096-17148 A00000/testTag I Ability onWindowStageCreate
03-17 21:01:11.754 3096-17148 A00000/testTag I Ability onForeground
03-17 21:01:11.892 3096-17148 A00000/testTag I Succeeded in loading the content.
/pages/Index
1080 x 2340 (Phone)
Second页面
Index页面传来的数据
返回