Harmony OS
入门系列课程 <快速上手>

快速掌握鸿蒙系统应用开发基础操作技巧

第 3 讲：通过Intent实现Ability之间数据传递

本讲案例应用场景

- 应用场景：用户登录、搜索、信息查看；

本讲内容

- 功能模块1：输入模块；
- 功能模块2：显示模块；
- 功能模块3：数据传递。

本讲目标

- 掌握Ability创建的方法；
- 掌握Button的使用方法；
- 掌握UI布局的设计方法
- 掌握实现Ability之间跳转的方法；
- 能够使用Intent实现数据传递。

上讲回顾【通过XML创建布局实现跑马灯】

步骤1：创建XML布局文件

步骤2：修改XML布局文件；

步骤3：代码中完成加载XML布局并开启滚动功能

上讲回顾【通过XML创建布局实现跑马灯】

操作演示

功能模块1【输入模块】

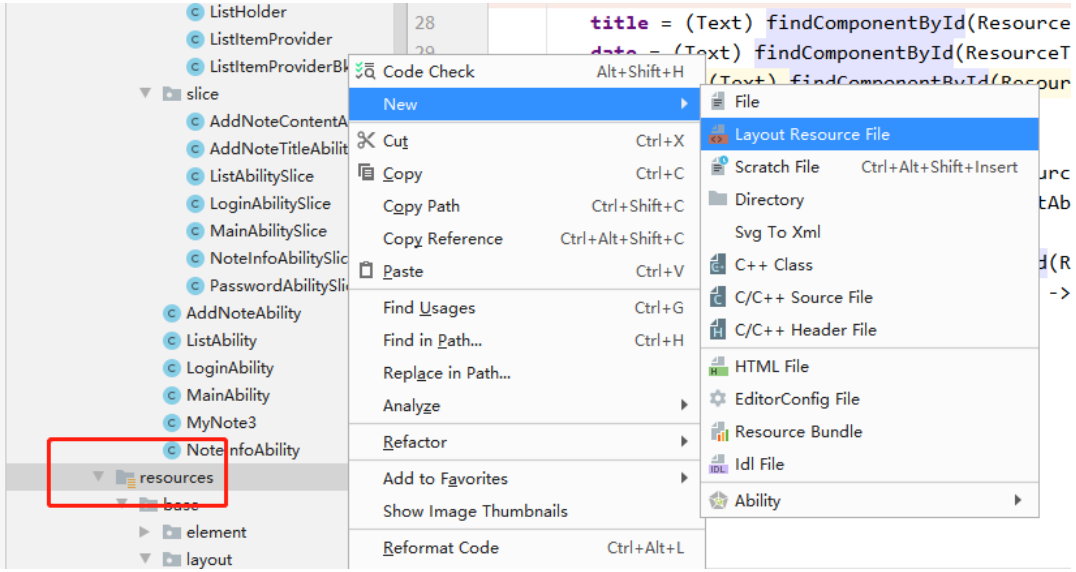
实现步骤：

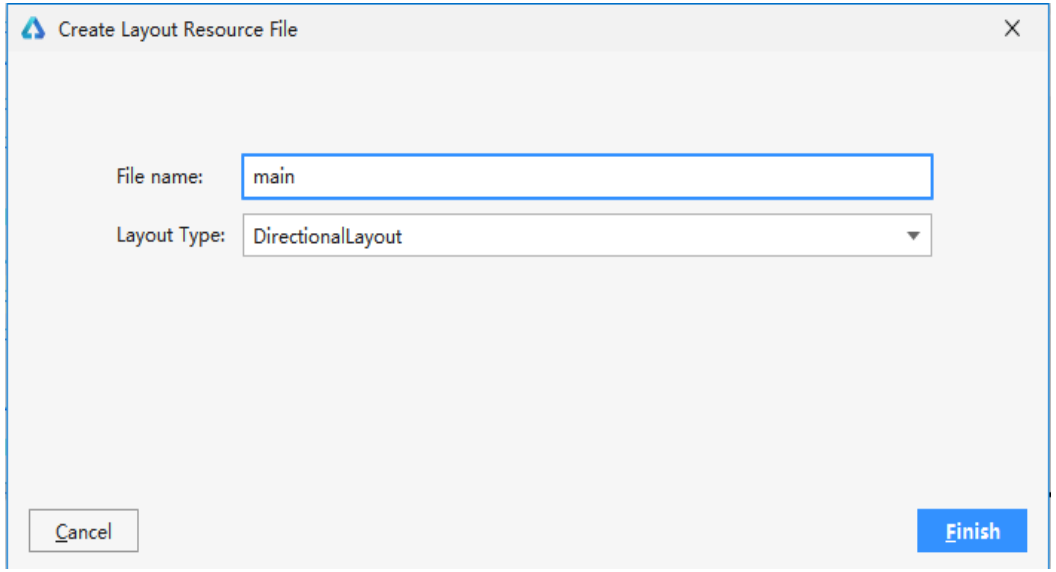
1. 创建工程及XML布局文件main.xml；
2. 修改main.xml布局文件完成UI;
3. 在MainAbilitySlice中加载布局文件main.xml

功能模块1【输入模块】

步骤1：创建工程及XML布局文件main.xml

1. 在DevEco Studio的“Project”窗口，打开“entry > src > main > ”，右键点击“resources”文件夹，选择“New > Layout Resource File”，命名为“main”





功能模块1【输入模块】

步骤2：修改main.xml布局文件完成UI;

<code/>

```
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout xmlns:ohos="http://schemas.huawei.com/res/ohos"
                    ohos:width="match_parent"
                    ohos:height="match_parent"
                    ohos:background_element="#eeeeee"
                    ohos:padding="30vp"
                    ohos:orientation="vertical">
```

功能模块1【输入模块】

步骤2：修改main.xml布局文件完成UI;

<code/>

```
<Text
        ohos:width="match_parent"
        ohos:height="match_content"
        ohos:text="请输入要查询的关键字"
        ohos:text_alignment="left"
        ohos:margin="20vp"
        ohos:text_size="50"
        ohos:text_color="black"/>
```

功能模块1【输入模块】

步骤2：修改main.xml布局文件完成UI;

<code/>

```
<TextField
        ohos:width="match_parent"
        ohos:height="60vp"
        ohos:text="鸿蒙"
        ohos:text_alignment="left"
        ohos:background_element="white"
        ohos:margin="10vp"
        ohos:padding="10vp"
        ohos:text_size="50"
        ohos:id="$+id:content"
        ohos:text_color="black"/>
```

功能模块1【输入模块】

步骤2：修改main.xml布局文件完成UI;

<code/>

```
<Button
        ohos:width="match_content"
        ohos:height="match_content"
        ohos:text="查询"
        ohos:text_alignment="center"
        ohos:layout_alignment="right"
        ohos:background_element="blue"
        ohos:padding="10vp"
        ohos:margin="20vp"
        ohos:id="$+id:send"
        ohos:text_size="50"
        ohos:text_color="white"/>  </DirectionalLayout>
```

功能模块1【输入模块】

步骤3：在MainAbilitySlice中加载布局文件main.xml

<code/>

```
public class MainAbilitySlice  extends AbilitySlice {


    @Override
    public void onStart(Intent intent) {
        super.onStart(intent);
        super.setUIContent(ResourceTable.Layout_main);
    }
}
```

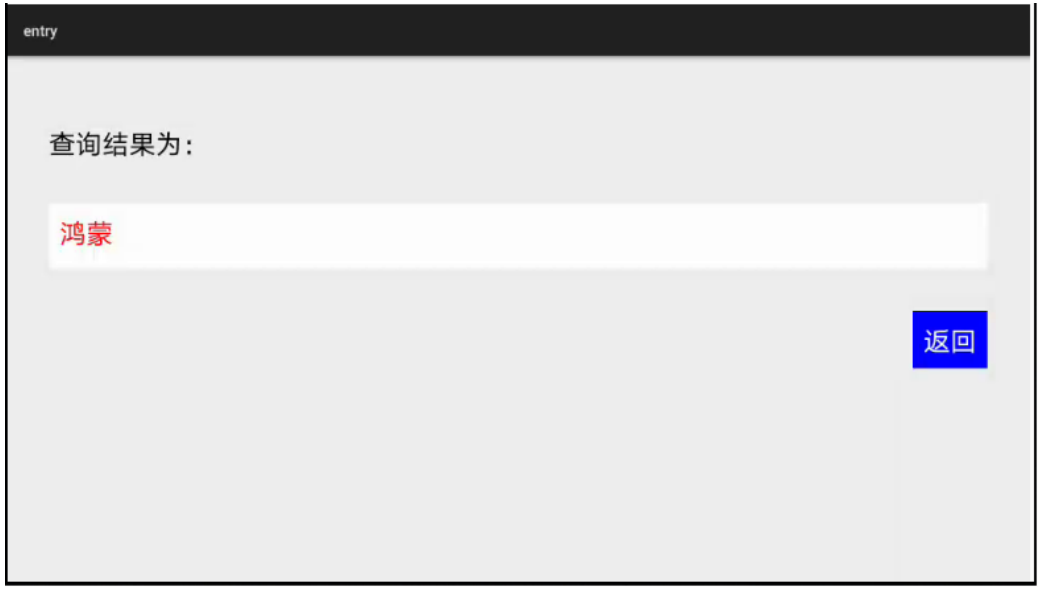
功能模块1【输入模块】

操作演示

功能模块2【显示模块】

实现步骤：

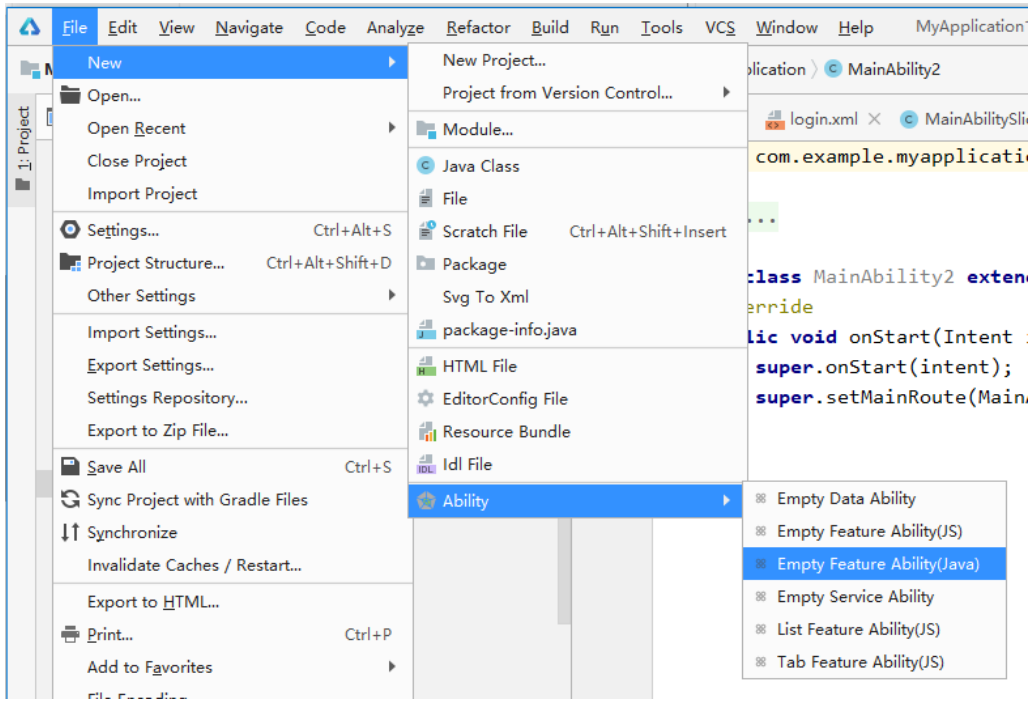
1. 创建ResultAbility,
2. 创建XML布局文件result.xml；
3. 修改result.xml布局文件完成UI;
4. 在ResultAbilitySlice中加载布局文件result.xml

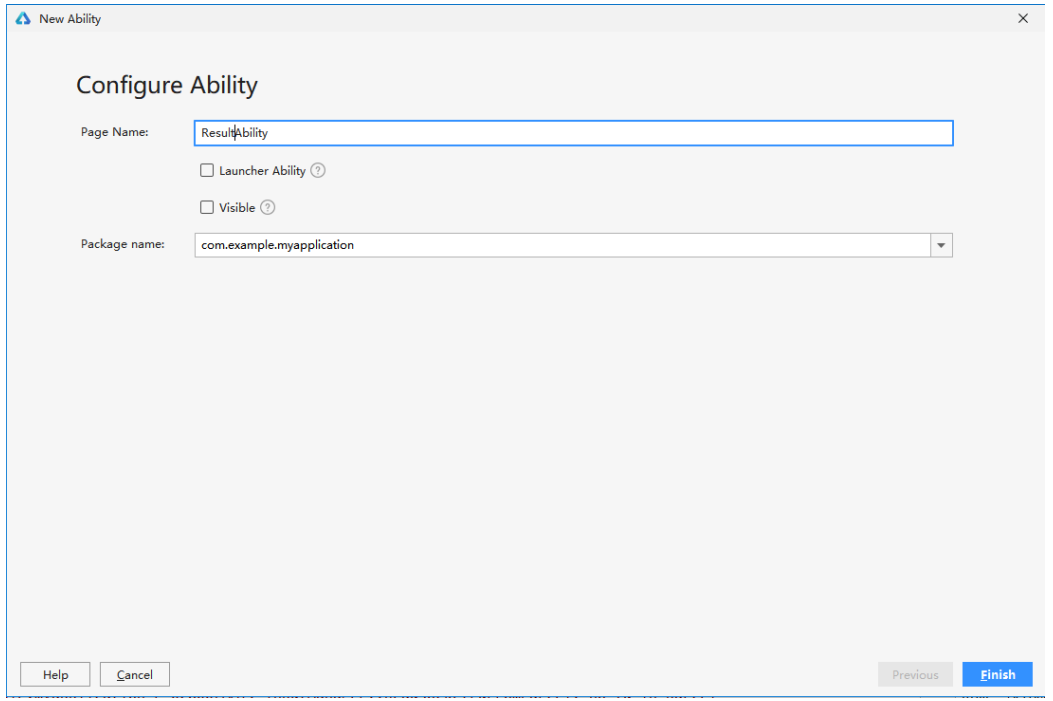


功能模块2【显示模块】

步骤1：创建ResultAbility

1. 在DevEco Studio的菜单栏，打开“File > New > Ability >Empty Feature Ability(java)

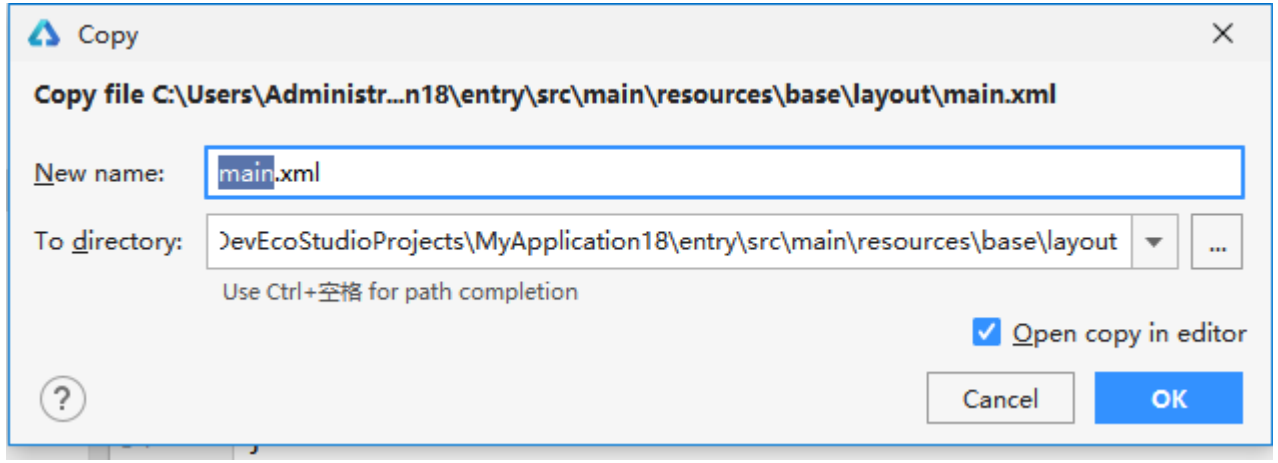




功能模块2【显示模块】

步骤2：创建XML布局文件result.xml；

1. 复制粘贴main.xml，并命名为result.xml

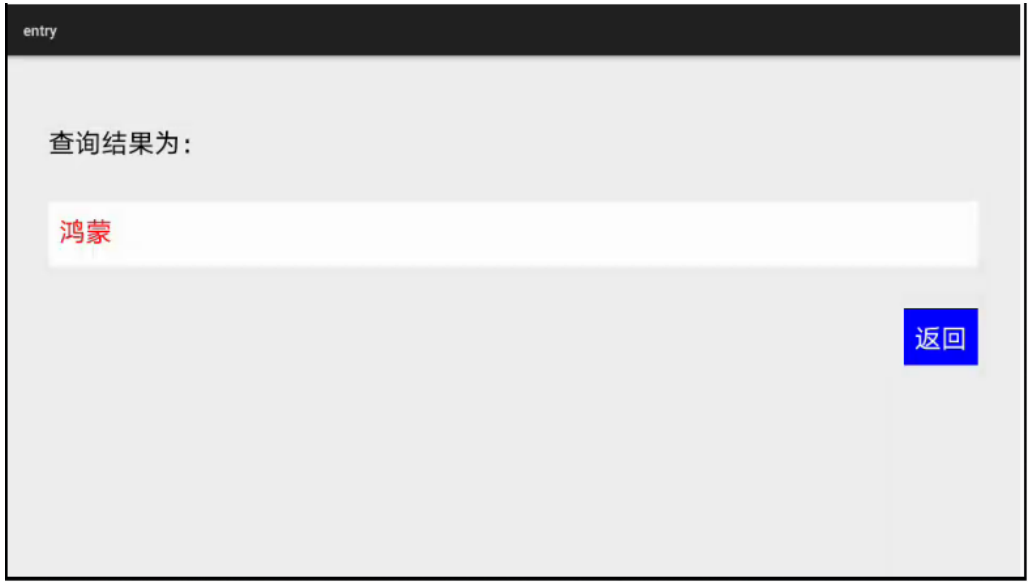


功能模块2【显示模块】

步骤3：修改result.xml布局文件完成UI

<code/>

```
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout
    xmlns:ohos="http://schemas.huawei.com/res/ohos"
            ohos:width="match_parent"
            ohos:height="match_parent"
            ohos:background_element="#eeeeee"
            ohos:padding="30vp"
            ohos:orientation="vertical">
```

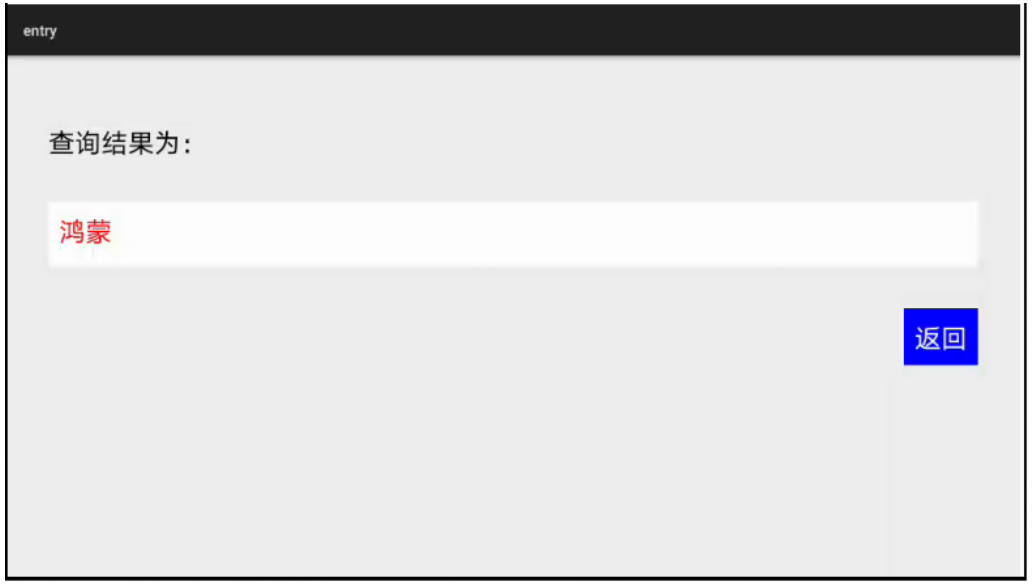


功能模块2【显示模块】

步骤3：修改result.xml布局文件完成UI

<code/>

```
<Text
    ohos:width="match_parent"
    ohos:height="match_content"
    ohos:text="查询结果为："
    ohos:text_alignment="left"
    ohos:margin="20vp"
    ohos:text_size="50"
    ohos:text_color="black"/>
```

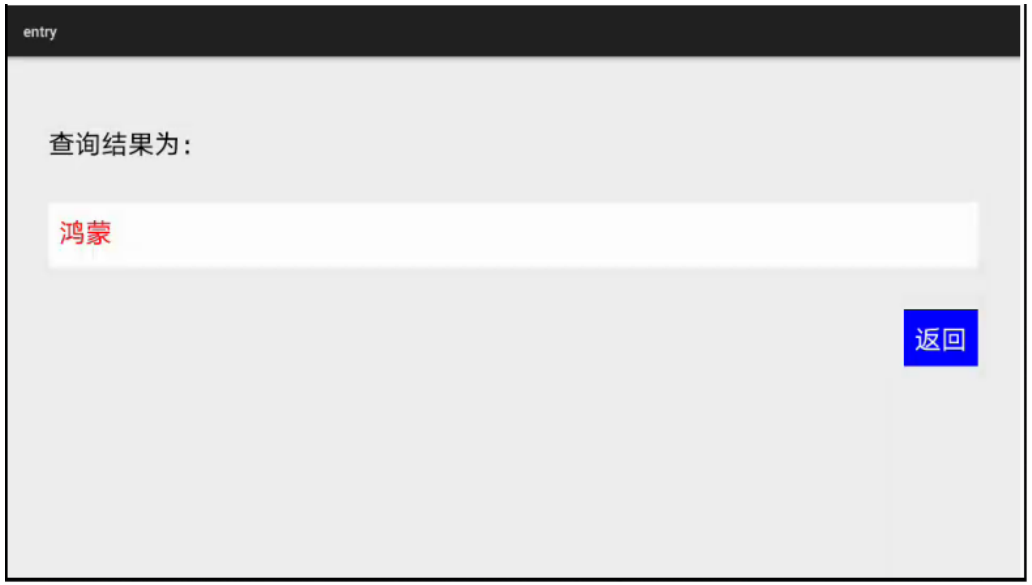


功能模块2【显示模块】

步骤3：修改result.xml布局文件完成UI

<code/>

```
<Text
    ohos:width="match_parent"
    ohos:height="60vp"
    ohos:text_color="red"
    ohos:background_element="white"
    ohos:padding="10vp"
    ohos:text_size="50"
    ohos:text_alignment="left"
    ohos:text="鸿蒙"
    ohos:id="$+id:text"
    ohos:margin="20vp"    />
```

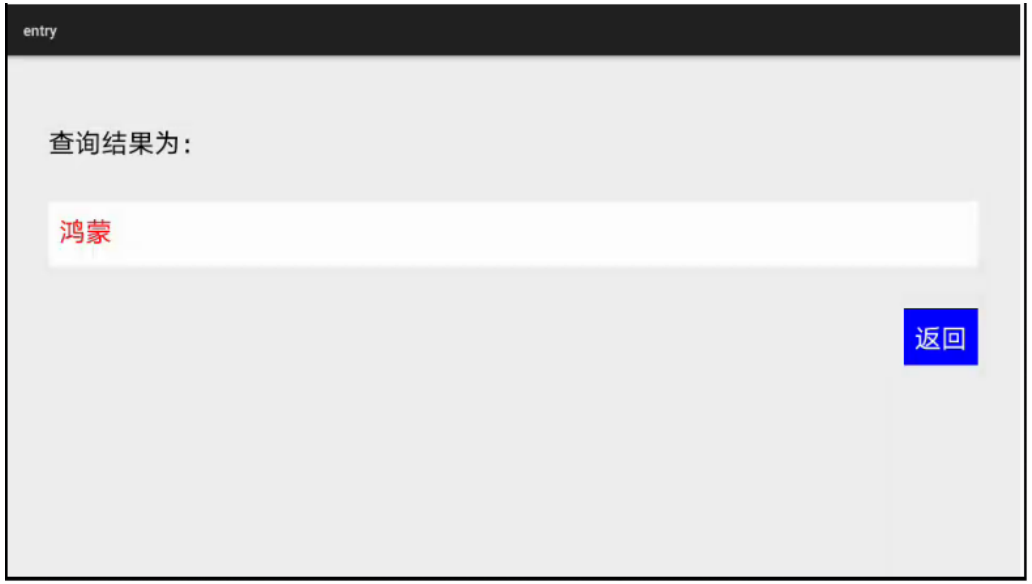


功能模块2【显示模块】

步骤3：修改result.xml布局文件完成UI

<code/>

```
<Button
    ohos:width="match_content"
    ohos:height="match_content"
    ohos:padding="10vp"
    ohos:margin="20vp"
    ohos:id="$+id:send"
    ohos:text="返回"
    ohos:background_element="blue"
    ohos:text_alignment="center"
    ohos:text_size="50"
    ohos:text_color="white"
    ohos:layout_alignment="right"    />
```



功能模块2【显示模块】

步骤4：在ResultAbilitySlice中加载布局文件result.xml

<code/>

```
public class ResultAbilitySlice extends AbilitySlice {


    @Override
    public void onStart(Intent intent) {
        super.onStart(intent);
        super.setUIContent(ResourceTable.Layout_result);
    }
}
```

功能模块2【显示模块】

操作演示

功能模块3【数据传递】

实现步骤：

1. 在MainAbility中获取UI组件
2. 给按钮添加点击事件
3. 完成启动跳转ResultAbility功能
4. 在Intent中添加数据；
5. 在ResultAbility中获取并显示数据
6. 完成返回功能

功能模块3【数据传递】

步骤1：在MainAbility中获取UI组件

<code/>

```
//1、在MainAbilitySlice中定义成员变量
Button send;
TextField content;

//2、在onStart方法中的下面通过id找到布局中的组件
send = (Button) findComponentById(ResourceTable.Id_send);
content = (TextField) findComponentById(ResourceTable.Id_content);
```

功能模块3【数据传递】

步骤2：给按钮添加点击事件

<code/>

```
//给Button组件添加点击事件
send .setClickedListener(new Component.ClickedListener() {
    //重写onClikc方法
    @Override
    public void onClick(Component component) {


      }
);
```

功能模块3【数据传递】

步骤3：完成启动跳转ResultAbility功能

<code/>

```
//在onClick方法中添加如下代码，完成ResultAbility启动功
 //创建Intent对象，Intent是对象之间传递信息的载体，In
Intent intent1 = new Intent();
Operation operation = new Intent.OperationBuilder()
        .withBundleName("com.example.myapplication")
        .withAbilityName("com.example.myapplication.ResultAbility").build();
intent1.setOperation(operation);
startAbility(intent1);
```

表1 Intent的构成元素

属性	子属性	描述
Operation	Action	表示动作，通常使用系统预置Action，应用也可以自定义Action。例如IntentConstants.ACTION_HOME表示返回桌面动作。
	Entity	表示类别，通常使用系统预置Entity，应用也可以自定义Entity。例如Intent.ENTITY_HOME表示在桌面显示图标。
	Uri	表示Uri描述。如果在Intent中指定了Uri，则Intent将匹配指定的Uri信息，包括scheme, schemeSpecificPart, authority和path信息。
	Flags	表示处理Intent的方式。例如Intent.FLAG_ABILITY_CONTINUATION标记在本地的一个Ability是否可以迁移到远端设备继续运行。
	BundleName	表示包描述。如果在Intent中同时指定了BundleName和AbilityName，则Intent可以直接匹配到指定的Ability。
	AbilityName	表示待启动的Ability名称。如果在Intent中同时指定了BundleName和AbilityName，则Intent可以直接匹配到指定的Ability。
	DeviceId	表示运行指定Ability的设备ID。
Parameters	-	Parameters是一种支持自定义的数据结构，开发者可以通过Parameters传递某些请求所需的额外信息。

功能模块3【数据传递】

步骤4：在Intent中添加数据

<code/>

```
//在onClick方法中添加如下代码，完成ResultAbility启动功
 //创建Intent对象，Intent是对象之间传递信息的载体，In
Intent intent1 = new Intent();
Operation operation = new Intent.OperationBuilder()
        .withBundleName("com.example.myapplication")
        .withAbilityName("com.example.myapplication.ResultAbility").build();
intent1.setOperation(operation);
intent1.setParam("content",content.getText());//添加数据，content为key，
startAbility(intent1);
```

表1 Intent的构成元素

属性	子属性	描述
Operation	Action	表示动作，通常使用系统预置Action，应用也可以自定义Action。例如IntentConstants.ACTION_HOME表示返回桌面动作。
	Entity	表示类别，通常使用系统预置Entity，应用也可以自定义Entity。例如Intent.ENTITY_HOME表示在桌面显示图标。
	Uri	表示Uri描述。如果在Intent中指定了Uri，则Intent将匹配指定的Uri信息，包括scheme, schemeSpecificPart, authority和path信息。
	Flags	表示处理Intent的方式。例如Intent.FLAG_ABILITY_CONTINUATION标记在本地的一个Ability是否可以迁移到远端设备继续运行。
	BundleName	表示包描述。如果在Intent中同时指定了BundleName和AbilityName，则Intent可以直接匹配到指定的Ability。
	AbilityName	表示待启动的Ability名称。如果在Intent中同时指定了BundleName和AbilityName，则Intent可以直接匹配到指定的Ability。
	DeviceId	表示运行指定Ability的设备ID。
Parameters	-	Parameters是一种支持自定义的数据结构，开发者可以通过Parameters传递某些请求所需的额外信息。

功能模块3【数据传递】

步骤5：在ResultAbility中获取并显示数据

<code/>

```
public class ResultAbilitySlice extends AbilitySlice {
    @Override
    public void onStart(Intent intent) {
        super.onStart(intent);
        setUIContent(ResourceTable.Layout_second);
        Text info = (Text) findComponentById(ResourceTable.Id_info);  //获取Text组件对象
        String s = intent.getStringParam("content");   //从Intent中获取并显示数据
        info.setText("来至MainAbility---->"+s);    //显示数据
    }
}
```

功能模块3【数据传递】

步骤6：完成返回功能

<code/>

```
public class ResultAbilitySlice extends AbilitySlice {
    public void onStart(Intent intent) {
            //省略部分代码.....
       //通过ID找到Button
        Button send = (Button) findComponentById(ResourceTable.Id_send);
        //注册点击事件
        send.setClickedListener(listener->getAbility().terminateAbility());

    }
}
```

功能模块3【数据传递】

操作演示

本讲小结

本讲所学功能模块有：

- 功能模块1：输入模块
- 功能模块2：显示模块
- 功能模块3：数据传递

任务挑战

自己手动完成本案例所有功能

THANKS

更多学习视频，关注宅客学院……