

JSUI框架入门

基于JavaUI框架
事件式编程范式
类似安卓应用开发
/
鸿蒙应用开发
\
基于JSUI框架
声明式编程范式
类似Web前端开发

1 实例与页面

创建工程时选择“Empty Feature Ability(JS)”模板。

例程：JSHello

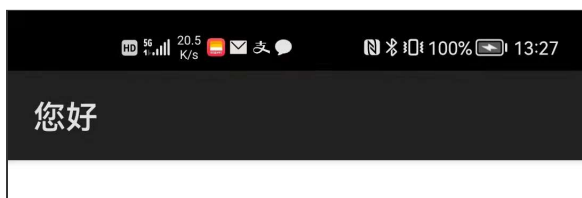
...\JSHello\entry\src\main\js\default\i18n\zh-CN.json

```
{
  "strings": {
    "hello": "您好",
    "world": "鸿蒙"
  }
}
```

...\JSHello\entry\src\main\js\default\i18n\en-US.json

```
{
  "strings": {
    "hello": "Hello",
    "world": "HarmonyOS"
  }
}
```

运行效果如下图所示：



您好 鸿蒙

Hello HarmonyOS

```

JSHello
|_entry          \
|_src            | 工程源码目录
|_main          /
|_config.json   - 工程配置文件
|
|_java          - Java目录
| |_com         \
|   |_minwei    | 应用包目录
|     |_jshello /
|       |_MainAbility.java - Ability类
|       |_MyApplication.java - 应用类
|
|_js            - JavaScript目录
| |_default     - JavaScript实例目录
|   |_app.js    - JavaScript实例入口
|   |
|   |_common    - 实例公共目录
|   | |_images  - 实例图像目录
|   |   |_bg-tv.jpg - 图像
|   |   |_Wallpaper.png - 图像
|   |
|   |_i18n      - 国际化目录
|   | |_en-US.json - 英文字符串
|   | |_zh-CN.json - 中文字符串
|   |
|   |_pages     - 实例页面目录
|   | |_index   - 页面目录
|   |   |_index.html - 页面结构
|   |   |_index.css - 页面外观
|   |   |_index.js - 页面行为
|   |
|   |_resources - 实例资源目录
|
|_resources     - 应用资源目录
|_base          - 标准资源目录
| |_element     - 文本资源目录
|   |_string.json - 应用字符串
|   |
|   |_media     - 媒体资源目录
|     |_icon.png - 应用图标
|
|_rawfile       - 原始资源目录

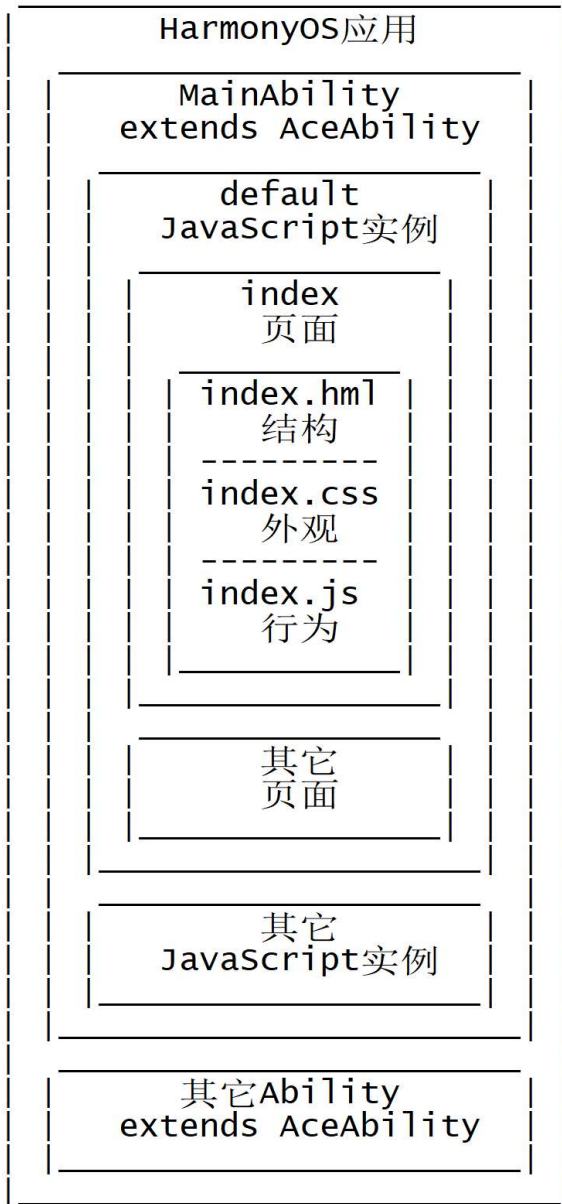
```

Java版:

```
public class MainAbility extends Ability { ... }
```

JS版:

```
public class MainAbility extends AceAbility { ... }
```



config.json:

```

{
  "module": {
    "abilities": [
      {
        "name": "com.minwei.jshello.MainAbility",
        ...
      },
      ...
    ],
    "js": [
      {
        "pages": [
          "pages/index/index" - index页面
          ... - 其它页面
        ],
        "name": "default",
        ...
      },
      ...
    ]
  }
}

```

\ MainAbility
 /
 \ 其它Ability
 /
 \ default
 / JavaScript实例
 \ 其它JavaScript实例
 /

- 一个HarmonyOS应用包含一到多个Ability， MainAbility是其中之一；
- 一个Ability包含一到多个JavaScript实例， default实例是其中之一；
- 一个JavaScript实例包含一到多个页面， index页面是其中之一；
- 一个页面由三个文件定义：
 - HML文件描述页面的结构——有什么？
 - CSS文件描述页面的外观——什么样？
 - JS文件描述页面的行为——干什么？

2 添加实例

除了默认的default实例以外，还可以添加新的JavaScript实例，方法有二：

- 创建新的Ability同时加载新的JavaScript实例；
- 让现有的Ability加载新的JavaScript实例。

2.1 创建新的Ability同时加载新的JavaScript实例

包名右键

New

Ability

Empty Page Ability(JS)

Page Name: SecondaryAbility
Launcher Ability: On
JS Component Name: secondary

2.2 让现有的Ability加载新的JavaScript实例

js右键

New

JS Component

JS Component Name: tertius

在需要加载tertius实例的Ability类的onStart()方法中调用:

```
super.setInstanceName("tertius");
```

例程: Instance

...\Instance\entry\src\main\java\com\minwei\instance\SecondaryAbility.java

```
public class SecondaryAbility extends AceAbility {  
    @Override  
    public void onStart(Intent intent) {  
        //setInstanceName("secondary");  
        setInstanceName("tertius");  
        ...  
    }  
    ...  
}
```

...\Instance\entry\src\main\js\default\i18n\zh-CN.json

```
{  
  "strings": {  
    "hello": "实例:",  
    "world": "default"  
  }  
}
```

...\Instance\entry\src\main\js\default\i18n\en-US.json

```
{  
  "strings": {  
    "hello": "Instance:",  
    "world": "default"  
  }  
}
```

...\Instance\entry\src\main\js\secondary\i18n\zh-CN.json

```
{
  "strings": {
    "hello": "实例:",
    "world": "secondary"
  }
}
```

...\Instance\entry\src\main\js\secondary\i18n\en-US.json

```
{
  "strings": {
    "hello": "Instance:",
    "world": "secondary"
  }
}
```

...\Instance\entry\src\main\js\tertius\i18n\zh-CN.json

```
{
  "strings": {
    "hello": "实例:",
    "world": "tertius"
  }
}
```

...\Instance\entry\src\main\js\tertius\i18n\en-US.json

```
{
  "strings": {
    "hello": "Instance:",
    "world": "tertius"
  }
}
```

运行效果如下图所示:



实例: default

实例: secondary

实例: tertius

3 初识页面

3.1 页面结构描述文件(.html)

```
<div class="container">
  <text class="title">
    {{ $t('strings.hello') }} {{ title }}
  </text>
</div>
```

- div: 块组件
- text: 文本框组件
- {{}}: 动态数据
- class: 类选择器

3.2 页面外观描述文件(.css)

```
.container {
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

.title {
  font-size: 40px;
}
```

- .xxx: 类选择器，引用自页面结构描述文件(.html)
- flex-direction: 布局方向，垂直(column)/水平(row)
- justify-content: 布局方向(主方向)上的对齐方式
- align-items: 布局正交方向(副方向)上的对齐方式

- font-size: 字体大小, 单位px表示像素

注意, 在config.json文件有关JavaScript实例的描述中:

```
"js": [  
  {  
    "pages": [  
      "pages/index/index"  
    ],  
    "name": "default",  
    "window": {  
      "designWidth": 720,      - 屏幕逻辑宽度  
      "autoDesignWidth": true - 屏幕逻辑宽度是否有效  
    }  
  }  
]
```

若autoDesignWidth为true, 即屏幕逻辑宽度有效, 则页面外观描述文件(.css)和页面行为描述文件(.js)中所有关于像素(px)的数值, 都指的是逻辑像素数:

设备像素数/逻辑像素数 = 屏幕设备宽度/屏幕逻辑宽度

如华为P40手机的屏幕宽度为1080像素, 按照如上配置, font-size为40px的文本实际高度为40x1080/720=60像素, 而在屏幕宽度为466像素的智能手表上, 其实际高度则只有40x466/720=25像素。

3.3 页面行为描述文件(.js)

```
export default {  
  data: {  
    title: ""  
  },  
  onInit() {  
    this.title = this.$t('strings.world');  
  }  
}
```

- title: 动态数据, 其值被页面结构描述文件(.html)中的“{}”引用
- onInit(): 生命周期方法, 创建页面时调用

4 页面跳转

在一个JavaScript实例内添加页面:

pages右键
New

JS Page

JS Page Name: secondary

config.json:

```
"js": [  
  {  
    "pages": [  
      "pages/index/index",  
      "pages/secondary/secondary"  
    ],  
    "name": "default",  
    ...  
  }  
]
```

```
import router from '@system.router';
```

```

|         |
|         |
|         |
|-----|
| index  | <- 显示
|_____|
|
| router.push({uri: "pages/secondary/secondary"}); // 入栈
| router.push({uri: "pages/secondary/tertius"}); // 入栈
v
|-----|
| tertius | <- 显示
|-----|
|secondary|
|-----|
| index  |
|_____|
|
| router.back(); // 出栈
v
|         |
|         |
|-----|
|secondary| <- 显示
|-----|
| index  |
|_____|
|
| router.replace({uri: "pages/secondary/tertius"}); // 先出栈再入栈
v
|         |
|         |
|-----|
| tertius | <- 显示
|-----|
| index  |
|_____|
|
| router.clear(); // 清除栈顶以下
v
|         |
|         |
|         |
|         |
|-----|
| tertius | <- 显示
|_____|

```

用户看见的永远都是栈顶页面，即当前页面。

跳转的同时还可传递参数：

```
route.push \  
    > ({uri: "pages/secondary/secondary",  
route.replace /    params: {param: this.string}});  
                |      |  
                参数名  参数值
```

目标页面直接从this中获取参数:

```
this.string = this.param;
```

例程: Jump

...\Jump\entry\src\main\js\default\pages\index\index.html

```
<div class="div">  
  <text class="txt">{{string}}</text>  
  <button class="btn" @click="onClick">跳转</button>  
</div>
```

...\Jump\entry\src\main\js\default\pages\index\index.css

```
.div {  
  flex-direction: column;  
  justify-content: center;  
  align-items: center  
}  
  
.txt {  
  height: 66px;  
  font-size: 22px;  
  color: #00a2e8;  
}  
  
.btn {  
  width: 120px;  
  height: 40px;  
  font-size: 24px;  
  background-color: #00a2e8;  
  color: #ffffff  
}
```

...\Jump\entry\src\main\js\default\pages\index\index.js

```
import router from '@system.router';

export default {
  data: {
    string: "鸿蒙初辟本无性，打破顽冥须悟空"
  },
  onClick() {
    router.push({uri: "pages/secondary/secondary",
      params: {param: this.string}});
  }
}
```

...\Jump\entry\src\main\js\default\pages\secondary\secondary.html

```
<div class="div">
  <text class="txt">{{string}}</text>
  <button class="btn" @click="onClick">返回</button>
</div>
```

...\Jump\entry\src\main\js\default\pages\secondary\secondary.css

```
.div {
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: #00a2e8;
}

.txt {
  height: 66px;
  font-size: 22px;
  color: #ffffff;
}

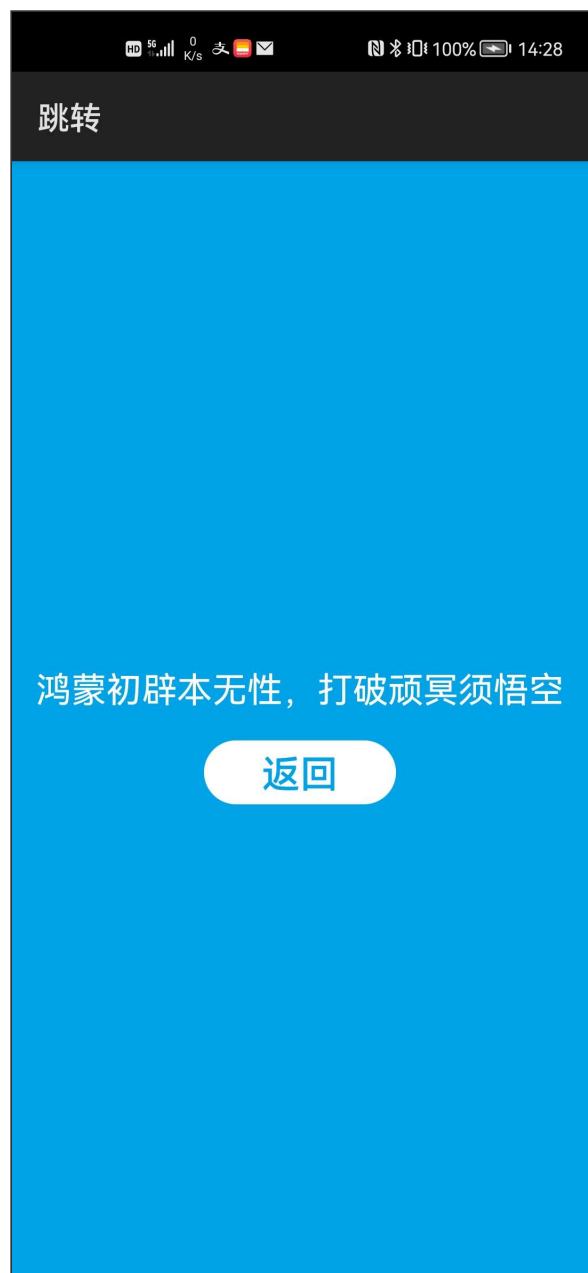
.btn {
  width: 120px;
  height: 40px;
  font-size: 24px;
  background-color: #ffffff;
  color: #00a2e8;
}
```

...\Jump\entry\src\main\js\default\pages\secondary\secondary.js

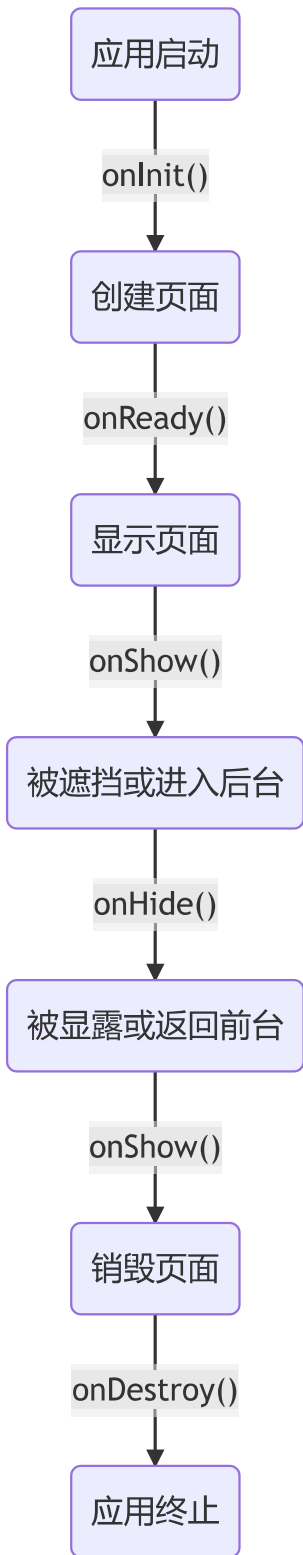
```
import router from '@system.router';

export default {
  data: {
    string: ""
  },
  onInit() {
    this.string = this.param;
  },
  onClick() {
    router.back();
  }
}
```

运行效果如下图所示：



5 生命周期



启动应用:



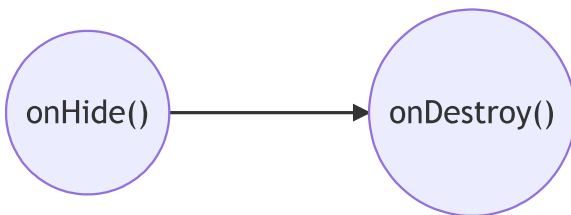
进入后台:



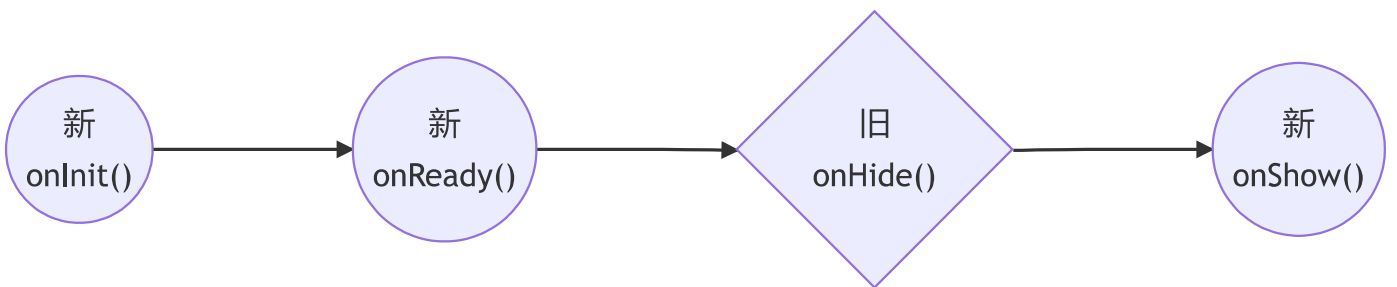
返回前台:



退出应用:



新页入栈:



新页出栈:



另外, 还有一个onBackPressed()方法, 会在通过手势或点击屏底按钮返回时被调用。

例程: Life

...\Life\entry\src\main\js\default\pages\index\index.html

```
<div class="div">
  <text class="txt">{{string}}</text>
  <button class="btn" @click="onClick">跳转</button>
</div>
```


...\Life\entry\src\main\js\default\pages\index\index.css

```
.div {
  flex-direction: column;
  justify-content: center;
  align-items: center
}

.txt {
  height: 66px;
  font-size: 22px;
  color: #22b14c;
}

.btn {
  width: 120px;
  height: 40px;
  font-size: 24px;
  background-color: #22b14c;
  color: #ffffff
}
```

...\Life\entry\src\main\js\default\pages\index\index.js

```
import router from '@system.router';

export default {
  data: {
    string: "我命在我，不属天地"
  },
  onInit() {
    console.log("index.onInit()");
  },
  onReady() {
    console.log("index.onReady()");
  },
  onShow() {
    console.log("index.onShow()");
  },
  onHide() {
    console.log("index.onHide()");
  },
  onDestroy() {
    console.log("index.onDestroy()");
  },
  onBackPress() {
    console.log("index.onBackPress()");
  },
  onClick() {
    router.push({uri: "pages/secondary/secondary",
      params: {param: this.string}});
  }
}
```

...\Life\entry\src\main\js\default\pages\secondary\secondary.html

```
<div class="div">
  <text class="txt">{{string}}</text>
  <button class="btn" @click="onClick">返回</button>
</div>
```

```
.div {
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: #22b14c;
}

.txt {
  height: 66px;
  font-size: 22px;
  color: #ffffff;
}

.btn {
  width: 120px;
  height: 40px;
  font-size: 24px;
  background-color: #ffffff;
  color: #22b14c;
}
```

..\Life\entry\src\main\js\default\pages\secondary\secondary.js

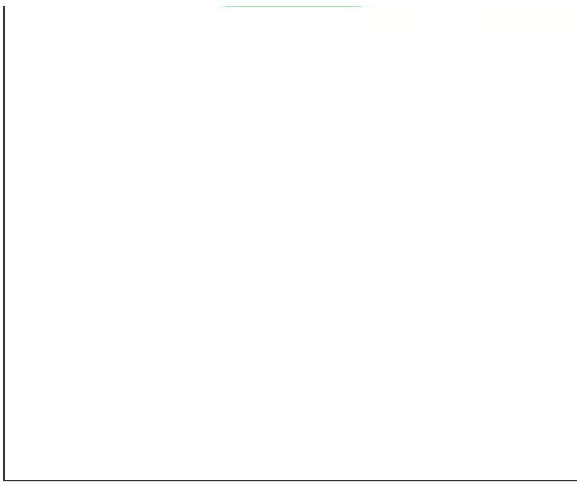
```
import router from '@system.router';

export default {
  data: {
    string: ""
  },
  onInit() {
    console.log("secondary.onInit()");

    this.string = this.param;
  },
  onReady() {
    console.log("secondary.onReady()");
  },
  onShow() {
    console.log("secondary.onShow()");
  },
  onHide() {
    console.log("secondary.onHide()");
  },
  onDestroy() {
    console.log("secondary.onDestroy()");
  },
  onBackPress() {
    console.log("secondary.onBackPress()");
  },
  onClick() {
    router.back();
  }
}
```

运行效果如下图所示：





```
Log - Life
Log: Hilog Fault Log
HUAWEI OCE-AN10 Show only selected application com.minwei.life (511) Debug app Log
content
10-14 15:16:04.411 511-4225/com.minwei.life I 03B00/JSApp: app Log: AceApplication onCreate
10-14 15:16:04.415 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onInit()
10-14 15:16:04.418 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onReady()
10-14 15:16:04.455 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onShow()
10-14 15:16:23.764 511-4225/com.minwei.life D 03B00/JSApp: app Log: secondary.onInit()
10-14 15:16:23.766 511-4225/com.minwei.life D 03B00/JSApp: app Log: secondary.onReady()
10-14 15:16:23.768 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onHide()
10-14 15:16:24.143 511-4225/com.minwei.life D 03B00/JSApp: app Log: secondary.onShow()
10-14 15:16:31.036 511-4225/com.minwei.life D 03B00/JSApp: app Log: secondary.onBackPressed()
10-14 15:16:31.045 511-4225/com.minwei.life D 03B00/JSApp: app Log: secondary.onHide()
10-14 15:16:31.412 511-4225/com.minwei.life D 03B00/JSApp: app Log: secondary.onDestroy()
10-14 15:16:31.418 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onShow()
10-14 15:16:34.502 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onBackPressed()
10-14 15:16:34.503 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onHide()
10-14 15:16:34.505 511-4225/com.minwei.life D 03B00/JSApp: app Log: index.onDestroy()
10-14 15:16:35.097 511-4225/com.minwei.life I 03B00/JSApp: app Log: AceApplication onDestroy
```

6 应用对象

每个JavaScript实例都有一个app.js文件，该文件包含了JavaScript实例的生命周期方法：

- onCreate(): 创建JavaScript实例时被调用；
- onDestroy(): 销毁JavaScript实例时被调用。

在app.js中还可以定义一些为该实例所有页面共享的全局性属性和方法。在从跳转的目标页面返回时，可以借助应用对象携带必要的返回信息。

页面js通过this.\$app访问这些标识符。

例程：App

...\App\entry\src\main\js\default\app.js

```
export default {
  data: {
    count: null
  },
  onCreate() {
    console.info('AceApplication onCreate');

    this.count = 0;
  },
  onDestroy() {
    console.info('AceApplication onDestroy');
  },
  getCount() {
    return this.count;
  },
  incCount() {
    ++this.count;
  }
};
```

...\App\entry\src\main\js\default\pages\index\index.html

```
<div class="div">
  <button class="btn" @click="onClick">{{count}}</button>
</div>
```

...\App\entry\src\main\js\default\pages\index\index.css

```
.div {
  flex-direction: column;
  justify-content: center;
  align-items: center
}

.btn {
  width: 150px;
  height: 50px;
  font-size: 30px;
  background-color: #ff7f27;
  color: #ffffff
}
```

...\App\entry\src\main\js\default\pages\index\index.js

```
import router from '@system.router';

export default {
  data: {
    count: null
  },
  onShow() {
    this.count = this.$app.getCount();
  },
  onClick() {
    router.push({uri: "pages/secondary/secondary"});

    this.$app.incCount();
  },
}
```

...\App\entry\src\main\js\default\pages\secondary\secondary.html

```
<div class="div">
  <button class="btn" @click="onClick">{{count}}</button>
</div>
```

...\App\entry\src\main\js\default\pages\secondary\secondary.css

```
.div {
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: #ff7f27;
}

.btn {
  width: 150px;
  height: 50px;
  font-size: 30px;
  background-color: #ffffff;
  color: #ff7f27;
}
```

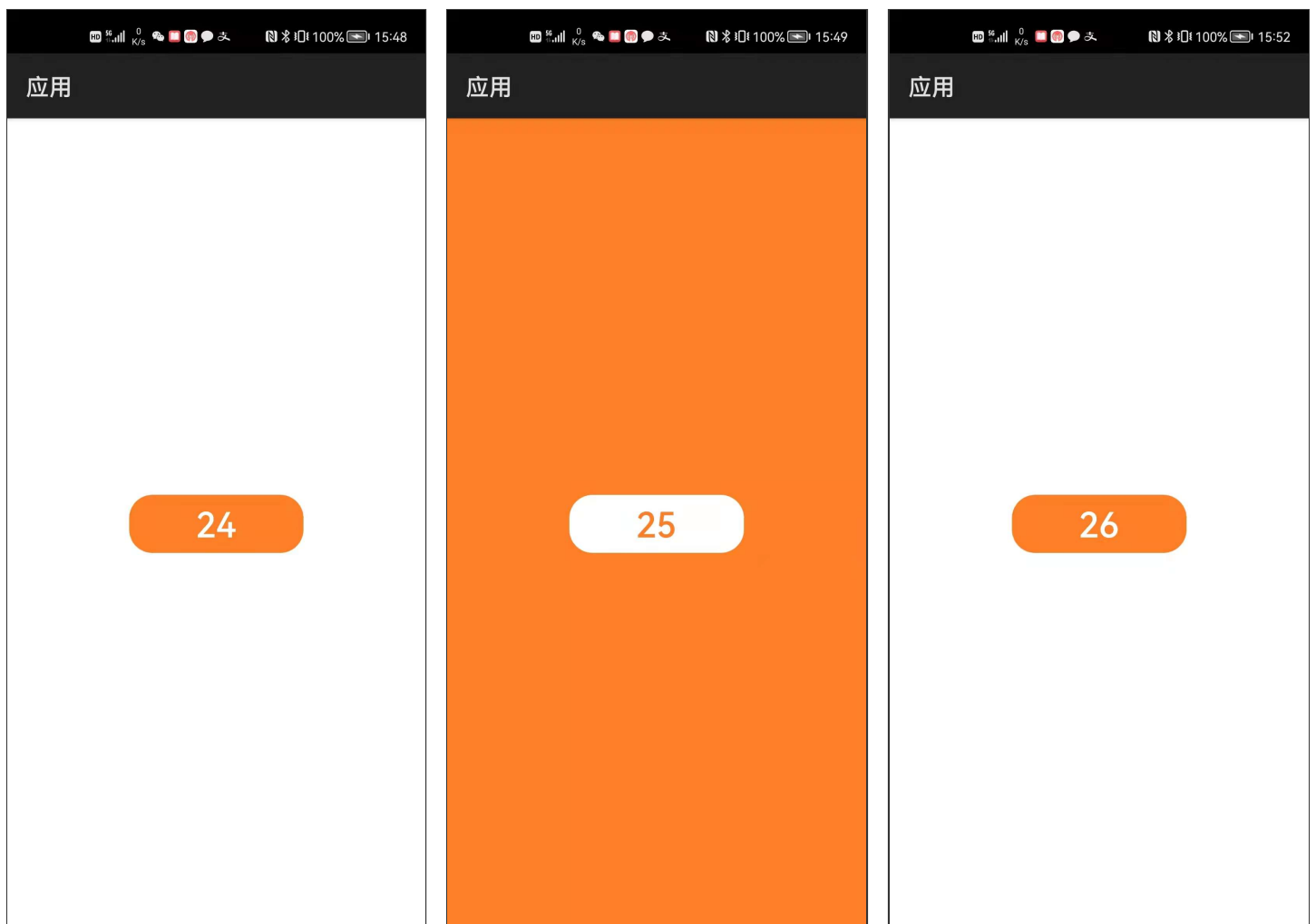
...\App\entry\src\main\js\default\pages\secondary\secondary.js

```
import router from '@system.router';

export default {
  data: {
    count: null
  },
  onInit() {
    this.count = this.$app.getCount();
  },
  onBackPress() {
    this.$app.incCount();
  },
  onClick() {
    router.back();

    this.$app.incCount();
  }
}
```

运行效果如下图所示：



更多精彩，敬请期待.....

达内集团C++教学部

2021年10月14日