1 请谈谈微信小程序主要目录和文件的作用？

* project.config.json 项目配置文件，用得最多的就是配置是否开启https校验；
* App.js   设置一些全局的基础数据等；
* App.json 底部tab, 标题栏和路由等设置；
* App.wxss 公共样式，引入iconfont等；
* pages 里面包含一个个具体的页面；
* index.json (配置当前页面标题和引入组件等)；
* index.wxml (页面结构)；
* index.wxss (页面样式表)；
* index.js (页面的逻辑，请求和数据处理等)；
* 1.1 简单描述下微信小程序的相关文件类型？
  + WXML （WeiXin Markup Language）是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构。内部主要是微信自己定义的一套组件。
  + WXSS (WeiXin Style Sheets)是一套样式语言，用于描述 WXML 的组件样式，
  + js 逻辑处理，网络请求
  + json 小程序设置，如页面注册，页面标题及tabBar。

app.json 必须要有这个文件，如果没有这个文件，项目无法运行，因为微信框架把这个作为配置文件入口，整个小程序的全局配置。包括页面注册，网络设置，以及小程序的window背景色，配置导航条样式，配置默认标题。

app.js 必须要有这个文件，没有也是会报错！但是这个文件创建一下就行 什么都不需要写以后我们可以在这个文件中监听并处理小程序的生命周期函数、声明全局变量。

app.wxss

#### 2 请谈谈wxml与标准的html的异同？

* 都是用来描述页面的结构；
* 都由标签、属性等构成；
* 标签名字不一样，且小程序标签更少，单一标签更多；
* 多了一些 wx:if 这样的属性以及 {{ }} 这样的表达式
* WXML仅能在微信小程序开发者工具中预览，而HTML可以在浏览器内预览
* 组件封装不同， WXML对组件进行了重新封装，
* 小程序运行在JS Core内，没有DOM树和window对象，小程序中无法使用window对象和document对象。

#### 3 请谈谈WXSS和CSS的异同？

* 都是用来描述页面的样子；
* WXSS 具有 CSS 大部分的特性，也做了一些扩充和修改；
* WXSS新增了尺寸单位，WXSS 在底层支持新的尺寸单位 rpx；
* WXSS 仅支持部分 CSS 选择器；
* WXSS 提供全局样式与局部样式

#### 4 你是怎么封装微信小程序的数据请求的？

* 在根目录下创建utils目录及api.js文件和apiConfig.js文件；
* 在apiConfig.js 封装基础的get, post 和 put， upload等请求方法，设置请求体，带上token和异常处理等；
* 在api中引入apiConfig.js封装好的请求方法，根据页面数据请求的urls, 设置对应的方法并导出；
* 在具体的页面中导入；

#### 5 小程序页面间有哪些传递数据的方法？

* 使用全局变量实现数据传递 在 app.js 文件中定义全局变量 globalData， 将需要存储的信息存放在里面

// app.js

App({

// 全局变量

globalData: {

userInfo: null

}

})

使用的时候，直接使用 getApp() 拿到存储的信息

* 使用 wx.navigateTo 与 wx.redirectTo 的时候，可以将部分数据放在 url 里面，并在新页面 onLoad 的时候初始化

//pageA.js

// Navigate

wx.navigateTo({

url: '../pageD/pageD?name=raymond&gender=male',

})

// Redirect

wx.redirectTo({

url: '../pageD/pageD?name=raymond&gender=male',

})

// pageB.js

...

Page({

onLoad: function(option){

console.log(option.name + 'is' + option.gender)

this.setData({

option: option

})

}

})

需要注意的问题：

wx.navigateTo 和 wx.redirectTo 不允许跳转到 tab 所包含的页面

onLoad 只执行一次

* 使用本地缓存 Storage 相关

#### 6 请谈谈小程序的双向绑定和vue的异同？

大体相同，但小程序直接this.data的属性是不可以同步到视图的，必须调用this.setData()方法！

#### 7 请谈谈小程序的生命周期函数？

* onLoad() 页面加载时触发，只会调用一次，可获取当前页面路径中的参数。
* onShow() 页面显示/切入前台时触发，一般用来发送数据请求；
* onReady() 页面初次渲染完成时触发, 只会调用一次，代表页面已可和视图层进行交互。
* onHide() 页面隐藏/切入后台时触发, 如底部 tab 切换到其他页面或小程序切入后台等。
* onUnload() 页面卸载时触发，如redirectTo或navigateBack到其他页面时。

#### 8 简述微信小程序原理？

* 小程序本质就是一个单页面应用，所有的页面渲染和事件处理，都在一个页面内进行，但又可以通过微信客户端调用原生的各种接口；
* 它的架构，是数据驱动的架构模式，它的UI和数据是分离的，所有的页面更新，都需要通过对数据的更改来实现；
* 它从技术讲和现有的前端开发差不多，采用JavaScript、WXML、WXSS三种技术进行开发；
* 功能可分为webview和appService两个部分；
* webview用来展现UI，appService有来处理业务逻辑、数据及接口调用；
* 两个部分在两个进程中运行，通过系统层JSBridge实现通信，实现UI的渲染、事件的处理等。
* 9. 你使用过哪些方法，来提高微信小程序的应用速度？
  + 提高页面加载速度
  + 用户行为预测
  + 减少默认data的大小
  + 组件化方案
* 10 小程序与原生App哪个好？
  + 小程序除了拥有公众号的低开发成本、低获客成本低以及无需下载等优势，在服务请求延时与用户使用体验是都得到了较大幅度  的提升，使得其能够承载跟复杂的服务功能以及使用户获得更好的用户体验。
* 11 简述微信小程序原理？
  + 微信小程序采用JavaScript、WXML、WXSS三种技术进行开发，从技术讲和现有的前端开发差不多，但深入挖掘的话却又有所不同。
  + JavaScript：首先JavaScript的代码是运行在微信App中的，并不是运行在浏览器中，因此一些H5技术的应用，需要微信App提供对应的API支持，而这限制住了H5技术的应用，且其不能称为严格的H5，可以称其为伪H5，同理，微信提供的独有的某些API，H5也不支持或支持的不是特别好。
  + WXML：WXML微信自己基于XML语法开发的，因此开发时，只能使用微信提供的现有标签，HTML的标签是无法使用的。
  + WXSS：WXSS具有CSS的大部分特性，但并不是所有的都支持，而且支持哪些，不支持哪些并没有详细的文档。
  + 微信的架构，是数据驱动的架构模式，它的UI和数据是分离的，所有的页面更新，都需要通过对数据的更改来实现。
  + 小程序分为两个部分webview和appService。其中webview主要用来展现UI，appService有来处理业务逻辑、数据及接口调用。它们在两个进程中运行，通过系统层JSBridge实现通信，实现UI的渲染、事件的处理
* 12 分析下微信小程序的优劣势？

优势：

无需下载，通过搜索和扫一扫就可以打开。

良好的用户体验：打开速度快。

开发成本要比App要低。

安卓上可以添加到桌面，与原生App差不多。

为用户提供良好的安全保障。小程序的发布，微信拥有一套严格的审查流程， 不能通过审查的小程序是无法发布到线上的。

劣势：

* + 限制较多。页面大小不能超过2M。不能打开超过9个层级的页面。
  + 样式单一。小程序的部分组件已经是成型的了，样式不可以修改。例如：幻灯片、导航。
  + 推广面窄，不能分享朋友圈，只能通过分享给朋友，附近小程序推广。其中附近小程序也受到微信的限制。
  + 依托于微信，无法开发后台管理功能。
  + 无需下载，通过搜索和扫一扫就可以打开。
  + 良好的用户体验：打开速度快。
  + 开发成本要比App要低。
  + 安卓上可以添加到桌面，与原生App差不多。
  + 为用户提供良好的安全保障。小程序的发布，微信拥有一套严格的审查流程， 不能通过审查的小程序是无法发布到线上的。
* 13 微信小程序与H5的区别？
  + 第一条是运行环境的不同
* 传统的HTML5的运行环境是浏览器，包括webview，而微信小程序的运行环境并非完整的浏览器，是微信开发团队基于浏览器内核完全重构的一个内置解析器，针对小程序专门做了优化，配合自己定义的开发语言标准，提升了小程序的性能。
  + 第二条是开发成本的不同
  + 只在微信中运行，所以不用再去顾虑浏览器兼容性，不用担心生产环境中出现不可预料的奇妙BUG
  + 第三条是获取系统级权限的不同
  + 系统级权限都可以和微信小程序无缝衔接
* 第四条便是应用在生产环境的运行流畅度
  + 长久以来，当HTML5应用面对复杂的业务逻辑或者丰富的页面交互时，它的体验总是不尽人意，需要不断的对项目优化来提升用户体验。但是由于微信小程序运行环境独立
  + 第四条便是应用在生产环境的运行流畅度

14 怎么解决小程序的异步请求问题？

回调函数方式：

app.js

success: function (info) {   
 that.apirtnCallback(info)  
}

index.js

onLoad: function () {  
 app.apirtnCallback = res => {  
 console.log(res)   
 }  
｝

小程序支持 ES6 语法

* 在返回成功的回调里面处理逻辑
* Promise 异步
* async/await
* 15 小程序的双向绑定和vue哪里不一样？

小程序直接this.data的属性是不可以同步到视图的，必须手动添加oninput调用this.setData({  
 noBind:true  
})

* 16 小程序的wxss和css有哪些不一样的地方？

WXSS 和 CSS 类似，不过在 CSS 的基础上做了一些补充和修改

尺寸单位 rpx： rpx 是响应式像素,可以根据屏幕宽度进行自适应。规定屏幕宽为 750rpx。如在 iPhone6 上，屏幕宽度为 375px，共有 750 个物理像素，则 750rpx = 375px = 750 物理像素

使用 @import 标识符来导入外联样式。@import 后跟需要导入的外联样式表的相对路径，用;表示语句结束

/\*\* index.wxss \*\*/

@import './base.wxss';

.container{

color: red;

}

* 17 webview中的页面怎么跳回小程序中？
* 答：首先在外部网页的HTML要引入最新版的jweixin-1.3.2.js：
* <script type="text/javascript"
* src="https://res.wx.qq.com/open/js/jweixin-1.3.2.js"></script>
* 然后

wx.miniProgram.navigateTo({ //如果跳转到tabBar中的页面，必须换成switchTo  
 url: '/pages/login/login'+'$params'  
 })

* 18 小程序关联微信公众号如何确定用户的唯一性？
  + 如果开发者拥有多个移动应用、网站应用、和公众帐号（包括小程序），可通过 unionid 来区分用户的唯一性，因为只要是同一个微信开放平台帐号下的移动应用、网站应用和公众帐号（包括小程序），用户的 unionid 是唯一的。换句话说，同一用户，对同一个微信开放平台下的不同应用，unionid 是相同的
* 19 如何实现下拉刷新？
* 首先在全局 config 中的 window 配置 enablePullDownRefresh
* 在 Page 中定义 onPullDownRefresh 钩子函数,到达下拉刷新条件后，该钩子函数执行，发起请求方法
* 请求返回后，调用 wx.stopPullDownRefresh 停止下拉刷新
* 20 使用webview直接加载要注意哪些事项？
  + 一, 必须要在小程序后台使用管理员添加业务域名；
  + 二、h5页面跳转至小程序的脚本必须是1.3.1以上；
  + 三、微信分享只可以都是小程序的主名称了，如果要自定义分享的内容，需小程序版本在1.7.1以上；
  + 四、h5的支付不可以是微信公众号的appid，必须是小程序的appid，而且用户的openid也必须是用户和小程序的。
* 21 小程序调用后台接口遇到哪些问题？

数据的大小有限制，超过范围会直接导致整个小程序崩溃，除非重启小程序；

* + 小程序不可以直接渲染文章内容页这类型的html文本内容，若需显示要借住插件，但插件渲染会导致页面加载变慢，所以最好在后台对文章内容的html进行过滤，后台直接处理批量替换p标签div标签为view标签，然后其它的标签让插件来做，减轻前端的时间。

22 webview的页面怎么跳转到小程序导航的页面？

小程序导航的页面可以通过switchTab，但默认情况是不会重新加载数据的。

若需加载新数据，则在success属性中加入以下代码即可：

success: function (e) {

var page = getCurrentPages().pop();

if (page == undefined || page == null) return;

page.onLoad();

}

webview的页面，则通过

wx.miniProgram.switchTab({

url: '/pages/index/index'

})

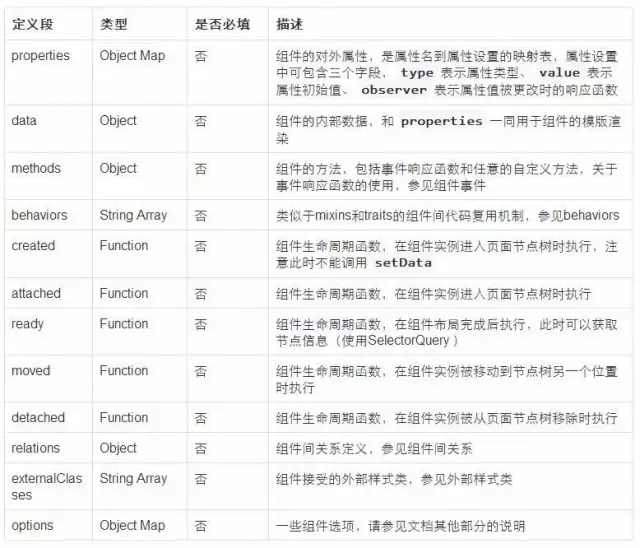
23 小程序和Vue写法的区别？

循环遍历的时候：小程序是wx:for="list"，而Vue是v-for="info in list"

调用data模型的时候：小程序是this.data.uinfo，而Vue是this.uinfo；给模型赋值也不一样，小程序是this.setData({uinfo:1})，而Vue是直接this.uinfo=1

* 24 为什么要对小程序进行组件模块化设计
  + 微信小程序有代码大小限制，通过测试最大为1M
  + 提高代码的复用率

25. 如何进行组件话模块化设计



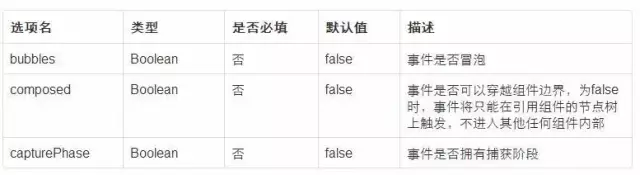
image

Component({  
 options: {  
 multipleSlots: true // 在组件定义时的选项中启用多slot支持  
 },  
 /\*\*  
 \* 组件的属性列表  
 \*/  
 properties: {  
 title: { // 属性名  
 type: String, // 类型（必填），目前接受的类型包括：String, Number, Boolean, Object, Array, null（表示任意类型）  
 value: '标题' // 属性初始值（可选），如果未指定则会根据类型选择一个  
 },  
 // 弹窗内容  
 content: {  
 type: String,  
 value: '内容'  
 },  
 // 弹窗取消按钮文字  
 btn\_no: {  
 type: String,  
 value: '取消'  
 },  
 // 弹窗确认按钮文字  
 btn\_ok: {  
 type: String,  
 value: '确定'  
 }   
 },  
   
 /\*\*  
 \* 组件的初始数据  
 \*/  
 data: {  
 flag: true,  
 },  
   
 /\*\*  
 \* 组件的方法列表  
 \*/  
 methods: {  
 //隐藏弹框  
 hidePopup: function () {  
 this.setData({  
 flag: !this.data.flag  
 })  
 },  
 //展示弹框  
 showPopup () {  
 this.setData({  
 flag: !this.data.flag  
 })  
 },  
 /\*  
 \* 内部私有方法建议以下划线开头  
 \* triggerEvent 用于触发事件  
 \*/  
 \_error () {  
 //触发取消回调  
 this.triggerEvent("error")  
 },  
 \_success () {  
 //触发成功回调  
 this.triggerEvent("success");  
 }  
 }  
 })

* + Component 构造器构造的组件也可以作为页面使用。
  + 使用 this.data 可以获取内部数据和属性值，但不要直接修改它们，应使用 setData 修改。
  + 生命周期函数无法在组件方法中通过 this 访问到。
  + 属性名应避免以 data 开头，即不要命名成 dataXyz 这样的形式，因为在 WXML 中， data-xyz="" 会被作为节点 dataset 来处理，而不是组件属性。
  + 在一个组件的定义和使用时，组件的属性名和data字段相互间都不能冲突（尽管它们位于不同的定义段中）。
  + Tips:
  + 通过WXML的import和include来使用文件模版
  + 使用WXSS的@import引用WXSS文件
  + 使用JS的require来引用JS文件

上边会用到一个 triggerEvent下面我们就来介绍下：

* 自定义组件触发事件时，需要使用 triggerEvent 方法，指定事件名、detail对象和事件选项。
* 触发事件的选项包括：



image

### 二：常见问题

**1.rpx（responsivepixel）**

微信小程序新定义了一个尺寸单位，可以适配不同分辨率的屏幕，它规定屏幕宽为750rpx，如在 iPhone6 上，屏幕宽度为375px，共有750个物理像素，则750rpx = 375px =750物理像素，1rpx = 0.5px = 1物理像素。

**2.40013错误**

在微信小程序刚出来的时候如果没输入AppID，提示这个信息就表示没有破解，但是现在官方软件更新可以选择无AppID开发，我们可以通过选择无AppID，即可解决此错误。建议安装官方开发工具。

**3.-4058错误**

微信小程序创建项目时选择无AppID，创建项目时会生成app.json，app.josn是程序启动最重要的文件，程序的页面注册，窗口设置，tab设置及网络请求时间设置都是在此文件下的。如果你创建的项目目录下没有app.json文件就会报下面的错误。

我们看到上面的错误信息中有个数字-4058，这应该是初入微信小程序遇到最多的错误了，这种一般都是文件缺失，后面有个path，可以对着该路径看看是否存在这个文件。造成这种错误的原因一般都是创建项目选择的目录不正确，或者在app.json注册了一个不存在的页面。

当然还有一种情况就是在app.json文件的pages注册的页面是没有创建的，或者你删除了某个页面，但是没有取消注册也会是-4058错误。

**4.Page注册错误**

这个错误可能很容易理解，页面注册错误。页面是通过Page对象来渲染的，每个页面对应的js文件必须要创建page，最简单的方式就是在js文件下写入Page({})，在page中有管理页面渲染的生命周期，以及数据处理，事件都在这完成。这个错误引起的原因一般都是刚创建页面，js文件还有有处理或者忘了处理。所以要养成创建页面的同时在js文件先创建Page的习惯.

**5.Page route错误**

字面意思就是页面路由错误，在微信中有两种路由方式一种是在wxml文件使用

如下代码：

**wxml文件：**

<navigatorurl="search/search">

<viewclass="serach\_view\_show" bindtap="bindtap"> 搜索</view>

</navigator>

**js文件事件处理函数：**

bindtap:function(event){

wx.navigateTo({

url: "search/search"

})

}

如果你这样写的话，恭喜你，你就会看到上面提示的错误，这是因为重复调用路由引起的，处理方法就是删除一个路由，删除<navigator />组件或者删除wx.navigateTo。除了上面说的可能导致路由错误外，还有一种情况，类似于下面的代码

<navigatorurl="search/search">

<navigatorurl="search/search">

<viewclass="serach\_view\_show" bindtap="bindtap"> 搜索</view>

</navigator>

</navigator>

这种也是不允许的，也就是说<navigator/>组件内部不能再嵌套<navigator/>组件。它只能是单层存在的。

**6.Do not have \* handler in currentpage.**

大概意思就是当前页面没有此处理，让确定是否已经定义，还指出了错误出现的可能位置pages/message/message,其实这种问题出现一般就是我们在wxml定义了一些处理事件，但是在js文件中没有实现这个时事件的处理方法，就会出现这个错误。那么我们按提示在js文件加上事件处理，如下代码，加上后就不会再有此错误提示。

  bindtap:function(event){

wx.navigateTo({

       url:"search/search"

})

  },

**7.tabBar设置不显示**

对于tabBar不显示，原因有很多，查找这个错误直接去app.json这个文件，最常见的也是刚学习微信小程序最容易犯的错误无外乎下面几种

**注册页面**即将页面写到app.json的pages字段中，如

"pages":[

    "pages/message/message",

    "pages/contact/contact",

    "pages/dynamic/dynamic",

    "pages/dynamic/music/music",

    "pages/index/index",

    "pages/logs/logs"

  ]

·  tabBar写法错误导致的不显示，将其中的大写字母B写成小写，导致tabBar不显示。

·  tabBar的list中没有写pagePath字段，或者pagePath中的页面没有注册

·  tabBar的list的pagePath指定的页面没有写在注册页面第一个。微信小程序的逻辑是"pages"中的第一个页面是首页，也就是程序启动后第一个显示的页面，如果tabBar的list的pagePath指定的页面都不是pages的第一个，当然也就不会显示tabBar了。

·  tabBar的数量低于两项或者高于五项，微信官方中明确规定tabBar的至少两项最多五项。超过或者少于都不会显示tabBar。

**8. 生命周期函数**

onLoad: 页面加载：一个页面只会调用一次，可以在 onLoad 中获取打开当前页面所调用的 query 参数。

onShow: 页面显示：每次打开页面都会调用一次。

onReady: 页面初次渲染完成： 一个页面只会调用一次，代表页面已经准备妥当，可以和视图层进行交互。对界面的设置如wx.setNavigationBarTitle请在onReady之后设置。详见生命周期

onHide: 页面隐藏：当navigateTo或底部tab切换时调用。

onUnload: 页面卸载：当redirectTo或navigateBack的时候调用。

**9.wx.navigateTo无法打开页面**

一个应用同时只能打开9个页面，当已经打开了9个页面之后，wx.navigateTo不能正常打开新页面。请避免多层级的交互方式，或者使用wx.redirectTo

**10.本地资源无法通过 css 获取**

background-image：可以使用网络图片，或者 base64，或者使用<image/>标签

**11.页面间数据传递**

微信小程序路由（页面跳转）是通过API wx.navigateTo或者wxml中<navigator/>组件实现的，不管哪种实现都会有一个重要的参数就是url，它指定了要跳转的页面，并且页面之间数据传递也是通过url来实现的，这个数据传递有点类似于我们使用的get网络请求，把参数都拼接在要跳转界面地址的后面并以“？”连接。然后将要传入的数据以键和值的形式追加在"?"后面，多个参数直接用"&"符合。如我们点击消息聊天记录，将列表上的数据传到下一个页面，可以这样写。

 <navigatorurl="/pages/dynamic/dynamic?title={{item.title}}&message={{item.message}}">

          <viewclass="item" >

             <viewclass="item-left">

                  <imagesrc="{{item.url}}" class="image"/>

             </view>

             <viewclass="item-middle">

                  <view>

                       <textclass="title">{{item.title}}</text>

                  </view>

                  <view>

                      <textclass="message">{{item.message}}</text>

                  </view>

            </view>

            <viewclass="item\_right">

               <view><text class="time">{{item.time}}</text></view>

                <viewclass="mark" wx:if="{{item.count>0}}"><textclass="text">{{item.count}}</text></view>

            </view>

         </view>

         <view  class="line"></view>

       </navigator>

 而数据接收是在js文件的page里接收的，page生命周期有一个onLoad函数，它就是做一些初始化数据的工作，onLoad函数有一个参数options，我们就可以通过key将数据获取，如下

Page({

  data:{

    // text:"这是一个页面"

    isHiddenToast:true

  }

  onLoad:function(options){

    // 页面初始化 options为页面跳转所带来的参数

    console.log(options.title)

    console.log(options.message)

  },

  onReady:function(){

    // 页面渲染完成

  },

  onShow:function(){

    // 页面显示

  },

  onHide:function(){

    // 页面隐藏

  },

  onUnload:function(){

    // 页面关闭

  },

  bindtap:function(event){

wx.navigateTo({

  url: "/pages/message/search/search"

})

  },

})

##### 12 bindtap和catchtap的区别是什么

相同点：首先他们都是作为点击事件函数，就是点击时触发。在这个作用上他们是一样的，可以不做区分

不同点：他们的不同点主要是bindtap是不会阻止冒泡事件的，catchtap是阻值冒泡的

##### 13 简述下 `wx.navigateTo()`, `wx.redirectTo()`, `wx.switchTab()`, `wx.navigateBack()`, `wx.reLaunch()`的区别

* wx.navigateTo()：保留当前页面，跳转到应用内的某个页面。但是不能跳到 tabbar 页面
* wx.redirectTo()：关闭当前页面，跳转到应用内的某个页面。但是不允许跳转到 tabbar 页面
* wx.switchTab()：跳转到 abBar 页面，并关闭其他所有非 tabBar 页面
* wx.navigateBack()关闭当前页面，返回上一页面或多级页面。可通过 getCurrentPages() 获取当前的页面栈，决定需要返回几层
* wx.reLaunch()：关闭所有页面，打开到应用内的某个页面