

02

OPEN ORIENTED

凹凸实验室

走进微信小程序（二）

实战篇

1

创建

2

配置

3

构建

4

开发

5

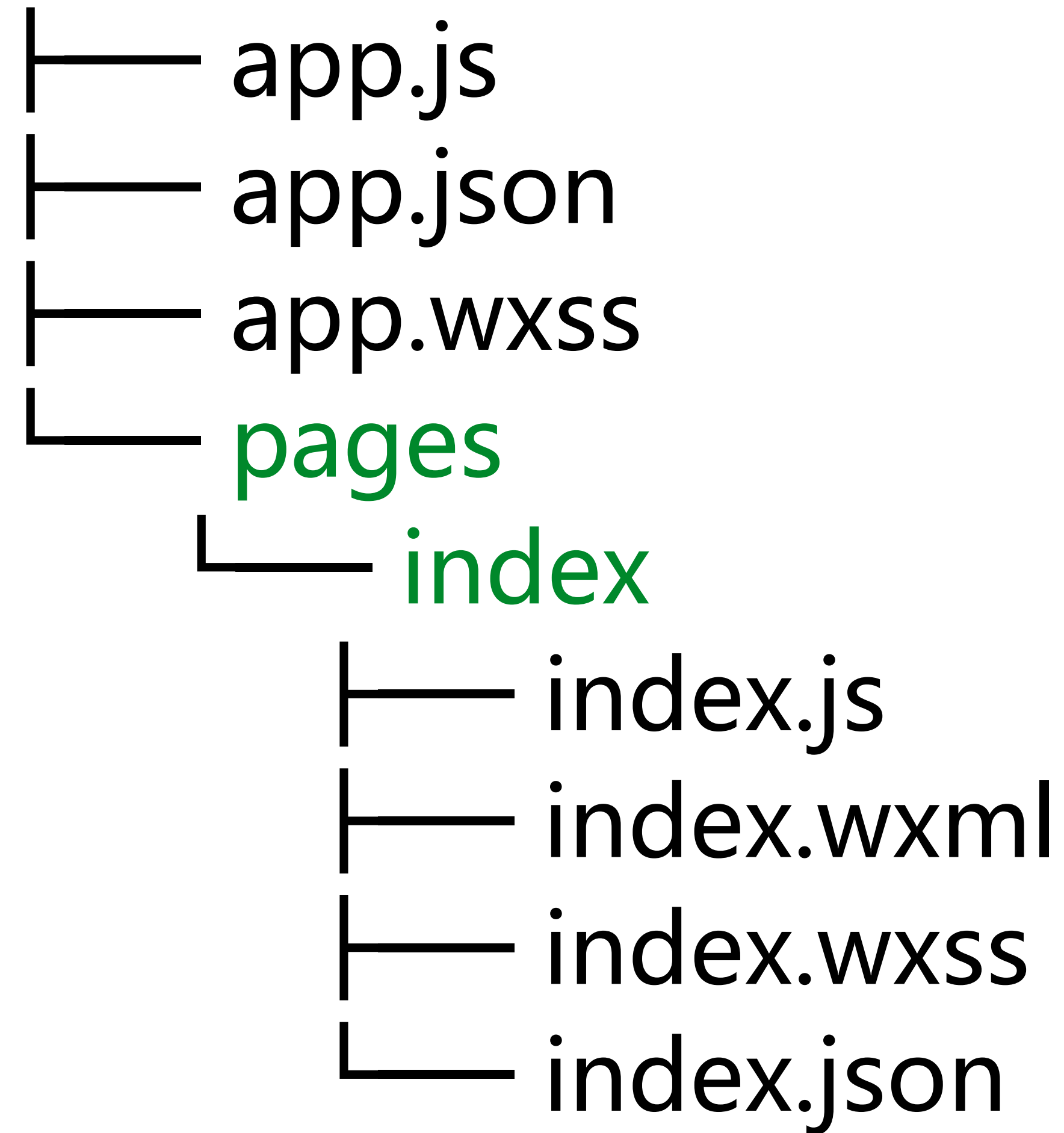
调试



1

创建

lesson2



2

配置

app.json全局配置

- pages
- window
- tabBar
- networkTimeout
- debug

window

02



navigationBar :

背景色、文案、字体颜色

background :

背景色、下拉文案、loading 图标

其它 :

是否开启下拉刷新

tabBar

02



数量：

2 ~ 5

样式：

- tab 普通样式
- tab 选中态样式
- tabBar 样式

page.json页面配置

- app.json 中 window 配置项
- disableScroll

- 注册小程序实例：

```
// app.js
```

```
App({ });
```

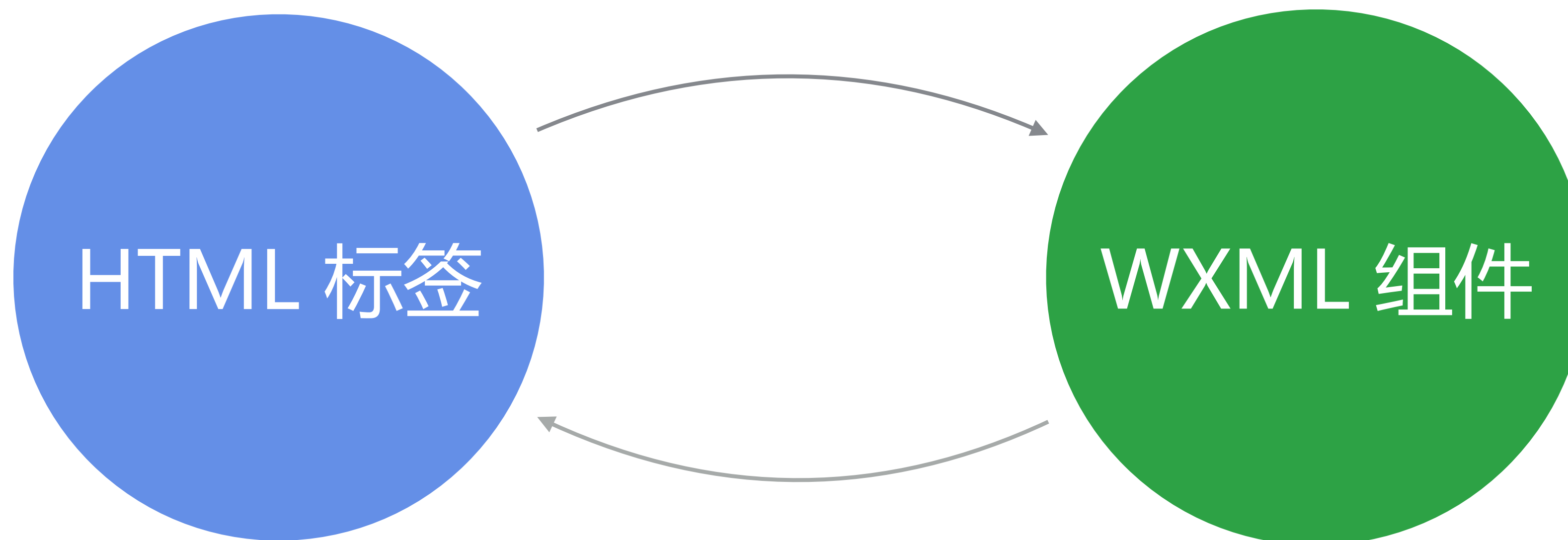
- 注册页面实例：

```
// page.js
```

```
Page({ });
```

3

构建




```
<template name="my_tpl">  
  <view>hello</view>  
</template>
```

<!-- 引用 -->

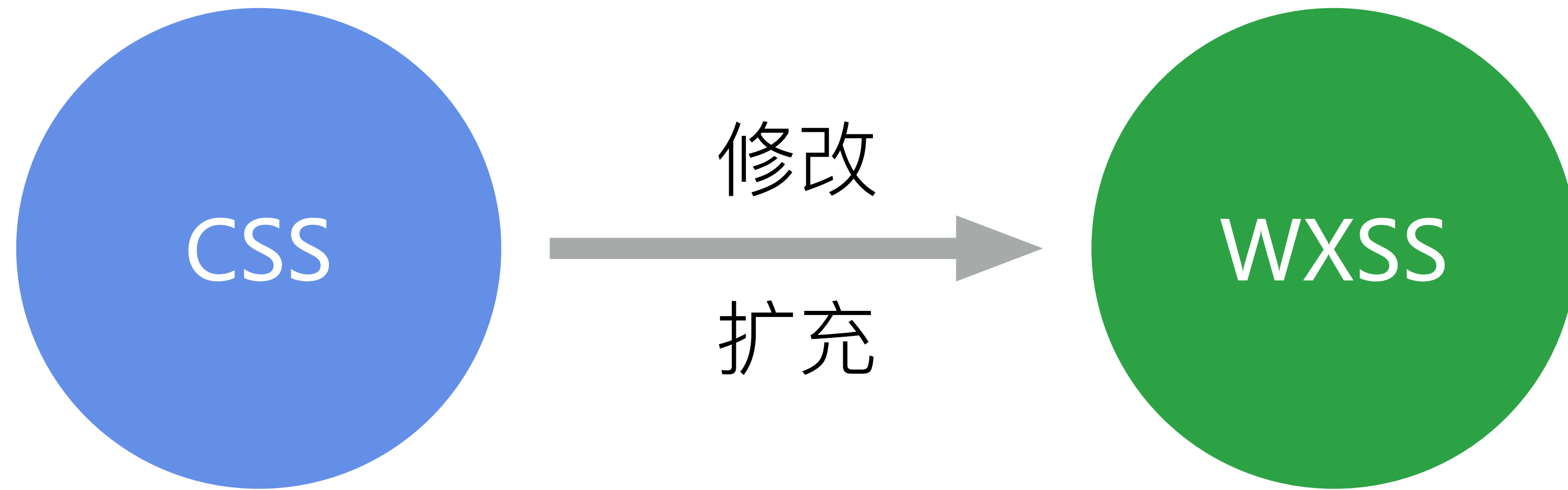
```
<import src="my_tpl.wxml">
```

<!-- 使用 -->

```
<template is="my_tpl" data="{{ ...data }}" />
```

<!-- 或 -->

```
<template is="my_tpl" data="{{ a: data1, b:data2 }}" />
```



WXSS 导入样式

```
@import "../my_tpl.wxss";
```


4

开发

```
// app.js
```

```
App({ });
```

属性	类型	描述
onLaunch	Function	小程序初始化完成时触发。
onShow	Function	小程序启动，或从后台进入前台时触发。
onHide	Function	小程序从前台进入后台时触发。
onError	Function	小程序脚本错误、api 调用失败时触发。
其它	任意	

```
const app = getApp();  
console.log(app.globalData);
```



```
// page.js
```

```
Page({ });
```

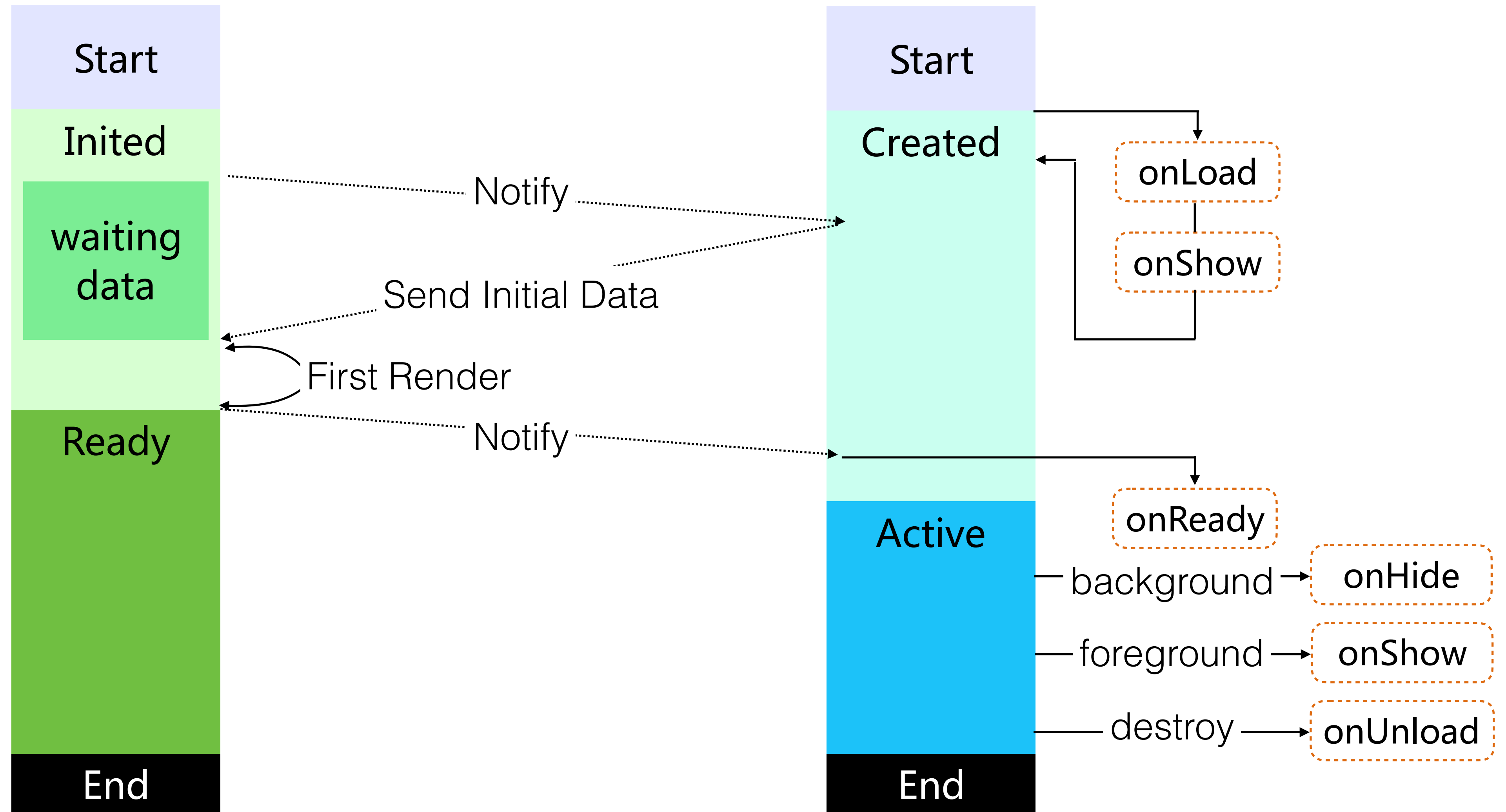
属性	类型	描述
data	Object	页面的初始数据。
生命周期函数	Function	
页面相关事件处理函数	Function	监听下拉刷新、触底事件、用户点击右上角分享。
其它	任意	

页面生命周期

02

View Thread

AppService Thread





Compare





```
let vm = new vm({  
  ...  
  data: {  
    a: 1  
  }  
});
```

修改数据 : `this.a = 99;`



```
Page({  
  ...  
  data: {  
    a: 1  
  }  
});
```

修改数据 : `this.setData({a: 99});`



```
let vm = new vm({  
  ...  
  computed: {  
    b: function() {  
      return this.a + 1;  
    }  
  }  
});
```



不支持



内容：

```
<div>{{ data }}</div>
```

HTML 片段：

```
<div>{{{ data }}}</div>
```



内容：

```
<view>{{ data }}</view>
```

HTML 片段：

不支持



标签属性：

```
<div v-bind:class="xx">hello</div>
```

控制属性：

```
<div v-if="showDiv">hello</div>
```



标签属性：

```
<view class="{{xx}}">hello</view>
```

控制属性：

```
<view wx:if="{{showDiv}}">hello</view>
```

{{ }} 里支持的 js 表达式



完全的 JavaScript 表达式支持



只支持几种简单的运算：

- 三元运算
- 算术运算
- 逻辑判断
- 字符串运算
- 等等

对 class 属性绑定的增强

22



对象语法、数组语法等：

```
<div :class="{ cur: isCur }">hello</div>
```

...



没对 class 的绑定做特殊处理：

```
<view class="{{ cur ? 'isCur' : null }}">hello</view>
```

```
<view class="{{ unfold }}" {{ fixed }}">hello</view>
```



```
<div v-if="type === 'A'">A</div>  
<div v-else-if="type === 'B'">B</div>  
<div v-else>Not A/B</div>  
  
<div v-show="isShow">hi</div>
```



```
<view wx:if="{{ type === 'A' }}">A</view>  
<view wx:elif="{{ type === 'B' }}">B</view>  
<view wx:else>Not A/B</view>  
  
<view hidden="{{ !isShow }}">hi</view>
```




```
<div v-for="(itemName, idx) in array"
  :key="itemName.id">
  {{ idx }}: {{ itemName.message }}
</div>
```



```
<view wx:for="{{ array }}"
      wx:for-item="itemName"
      wx:for-index="idx"Name"
      wx:key="id">
  {{ idx }}: {{ itemName.message }}
</view>
```



html :

```
<div @click="tapView">text</div>
```

js :

```
let vm = new vm({  
  methods: {  
    tapView: function(event) {}  
  }  
});
```



wxml :

```
<view bindtap="tapView"  
  data-arg="argument">  
  text</view>
```

js :

```
Page({  
  tapView: function(event) {}  
});
```



双向绑定：

```
<input v-model="message"/>
```



通过事件通知改写 data：

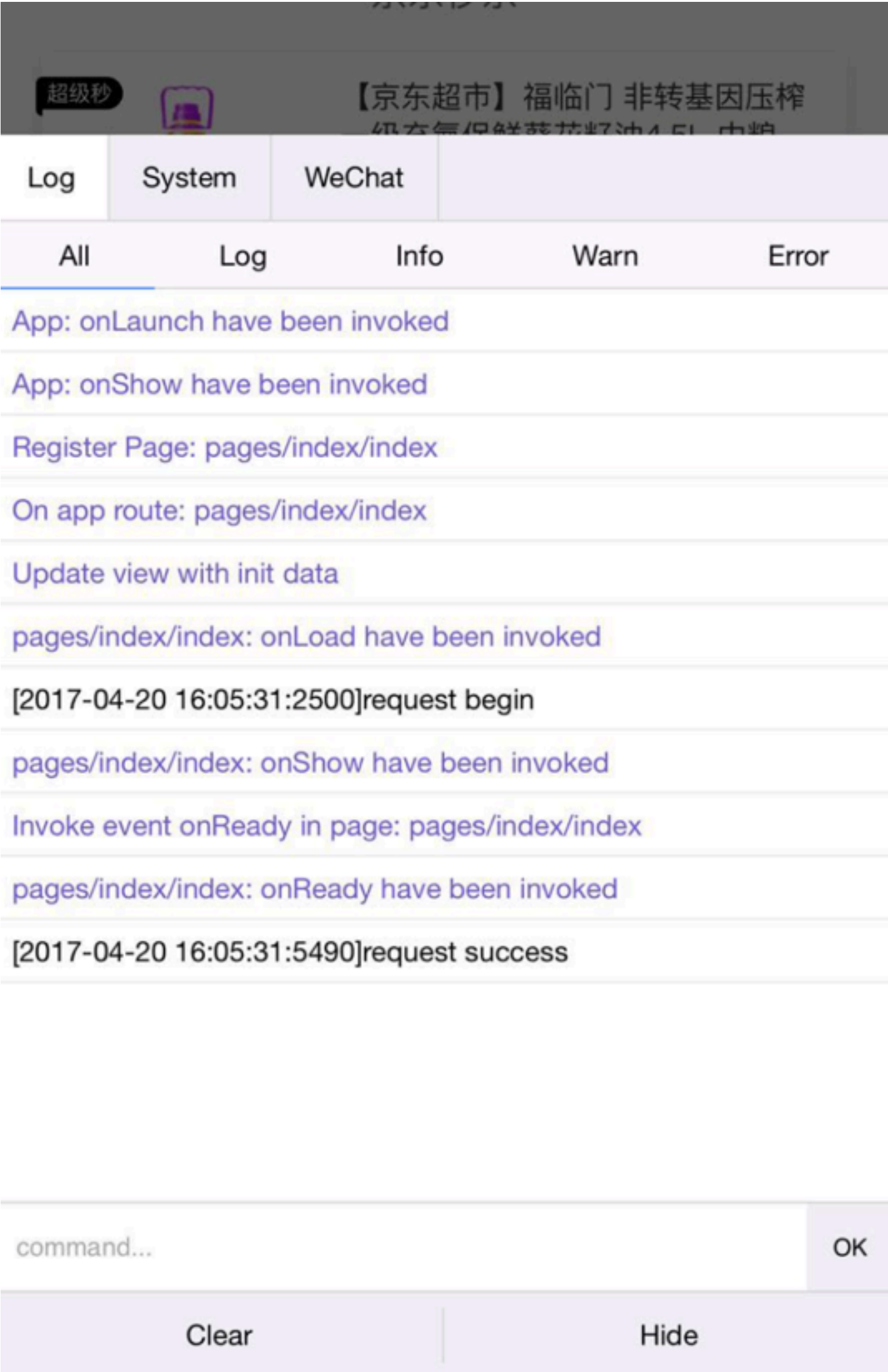
```
<input bindinput="bindInput">
```

```
Page({  
  ...  
  bindInput: function(e) {  
    this.setData({  
      message: e.detail.value  
    });  
  }  
});
```

5

调试

手机调试窗口



总结

学习成本



迁移成本



踩坑概率



作业

把自己手头上的项目的一个小模块迁移到小程序

可参考课程 demo：

http://git.jd.com/chenjiajian5/wxapp_lesson2_demo

Q&A