hell03W：iOSer 学习小程序第一天

教程详解: https://mp.weixin.qq.com/debug/wxadoc/dev/framework/MINA.html?t=20161102
晚上没事，打开看了一点点文档.

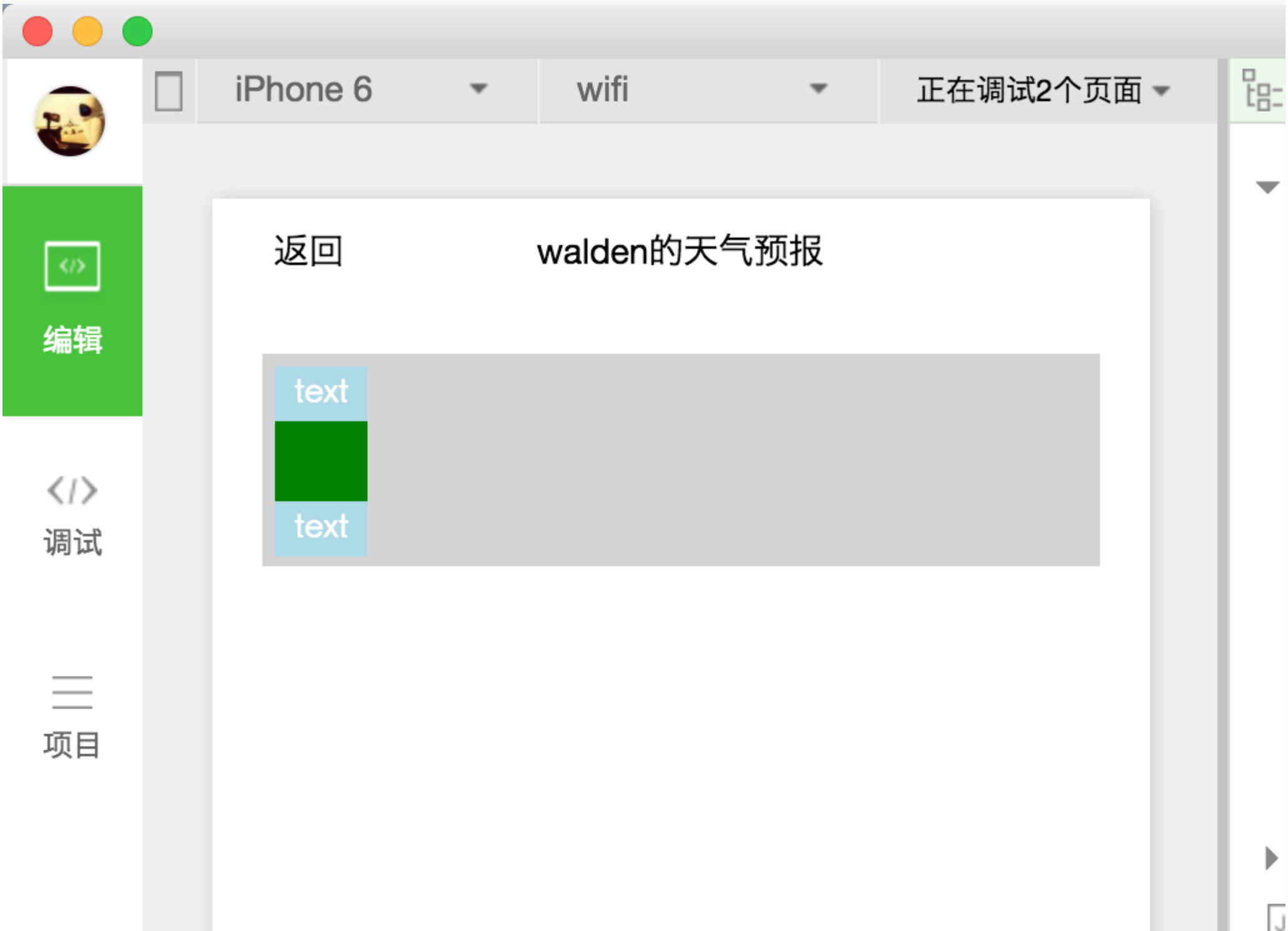
第一感觉，简单.

第二感觉，结构清晰.

小程序中，一个页面分为 4 部分组成. .js：页面逻辑处理和数据处理; .json：配置页面; .wxml：页面 UI 部分; wxss：页面样式部分.
作为一个 iOSer，作为一个习惯纯代码搞定一切的 iOSer，看到小程序这样的页面结构还是~，喜欢!
come on，明天花一天时间，学习小程序，测试框架和功能，最后做一个简单的天气预报小程序. 希望一天可以搞定吧.


●●○○○ 中国电信
22:04
上海市
Mostly Clear
Now 23 00 01 02 03 04
17° 16° 16° 15° 13° 13° 13
Sunday 22 14
Monday 23 13
Tuesday 14 10
Wednesday 13 10
Thursday 15 7

第三感觉，尼玛，这编辑器太烂了，不好用的很.....，不过刚开始第一版做成这样，可以原谅...


iPhone 6
wifi
正在调试2个页面
编辑
调试
项目
返回
walden的天气预报
text
text

1. 配置

https://mp.weixin.qq.com/debug/wxadoc/dev/framework/config.html?t=20161102

官方文档, 写的那么详细, 还有什么好记的 ...

2. 逻辑层

注册程序

app 的声明周期: onLaunch, onShow, onHide, 调用时刻 顾名思义吧.

```
//1.当小程序初始化完成时，会触发 onLaunch（全局只触发一次）
 onLaunch: function () {
   //调用 API 从本地缓存中获取数据
   var logs = wx.getStorageSync('logs') || []
   logs.unshift(Date.now())
   wx.setStorageSync('logs', logs)

   util.Logs("app on launch")
 },
//2.当小程序启动，或从后台进入前台显示，会触发 onShow
onShow: function () {
   util.Logs("app on show")
},
//3.当小程序从前台进入后台，会触发 onHide
onHide: function () {
  util.Logs("app on hide")
},
```

getApp(): 我们提供了全局的 getApp() 函数，可以获取到小程序实例。类似 iOS 中的 AppDelegate 类, 相当于是小程序的入口, getApp, 是我们获取这个类的实例的方法. 但是在 App 类中, 不能调用 getApp 函数, 首先当前已经是 App 类对象中, 可以调用其中任意方法(除生命周期相关方法和数据)和数据; 其次, ...,.

注册页面

一些生命周期函数，和数据对象，上拉刷新函数

```
//1.页面的初始数据，初始化数据将作为页面的第一次渲染。data 将会以 JSON 的形式由逻辑层传至渲染层，所以其数据必须是可以转成 JSON 的格式：字符串，数字，布尔值，对象，数组。
data: {
    motto: 'Hello World',
    userInfo: {}
  },

//2.生命周期函数--监听页面加载
//  一个页面只会调用一次。
// 接收页面参数可以获取 wx.navigateTo 和 wx.redirectTo 及<navigator/>中的 query。
  onLoad: function () {
    this.loadDataFromUserInfo()

    util.Logs("page on load")
  },
  //3.生命周期函数--监听页面初次渲染完成
  // 每次打开页面都会调用一次。
  onReady: function () {
    util.Logs("page on ready")
  },
  //4.生命周期函数--监听页面显示
  // 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
  onShow: function () {
    util.Logs("page on show")
  },
  //5.生命周期函数--监听页面隐藏
  // 当 navigateTo 或底部 tab 切换时调用。
  onHide: function () {
```

```
    util.Logs("page on hide")
  },
  //6.生命周期函数--监听页面卸载
  // 当 redirectTo 或 navigateBack 的时候调用。
  onUnload: function () {
    util.Logs("page on unload")
  },
  //7.监听用户下拉刷新事件。
  // 需要在 config 的 window 选项中开启 enablePullDownRefresh。
  // 当处理完数据刷新后，wx.stopPullDownRefresh 可以停止当前页面的下拉刷新。
  onPullDownRefresh: function () {
    util.Logs("pull down refresh")
  }
```

绑定事件

```
<view bindtap="viewTap"> click me </view>

Page({
  viewTap: function() {
    console.log('view tap')
  }
})
```

绑定数据 setData()

setData 函数用于将数据从逻辑层发送到视图层，同时改变对应的 this.data 的值。

注意：

1. 直接修改 this.data 无效，无法改变页面的状态，还会造成数据不一致。
2. 单次设置的数据不能超过 1024kB，请尽量避免一次设置过多的数据。

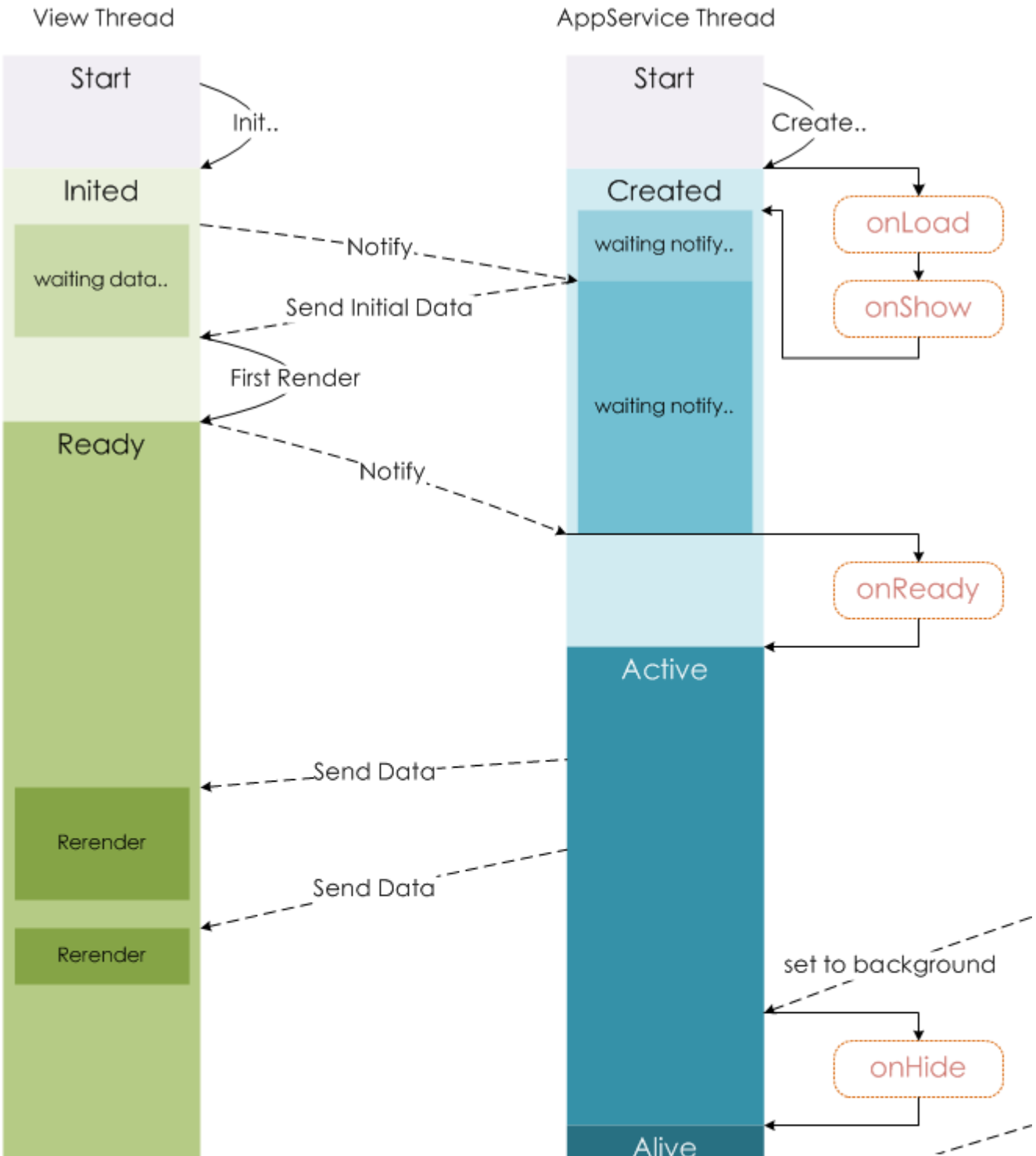
getCurrentPages()

getCurrentPages() 函数用于获取当前页面栈的实例，以数组形式按栈的顺序给出，第一个元素为首页，最后一个元素为当前页面。

注意：不要尝试修改页面栈，会导致路由以及页面状态错误。

pages 生命周期示意图

下图来自微信小程序开发文档....


View Thread
AppService Thread
Start
Init..
Inited
waiting data..
Notify
Send Initial Data
First Render
Ready
Notify
Rerender
Rerender
Send Data
Send Data
Start
Create..
Created
waiting notify..
waiting notify..
onLoad
onShow
onReady
Active
set to background
onHide
Alive

模块化

文件作用域

在 JavaScript 文件中声明的变量和函数只在该文件中有效；不同的文件中可以声明相同名字的变量和函数，不会互相影响。

通过全局函数 getApp() 可以获取全局的应用实例，如果需要全局的数据可以在 App() 中设置

模块化

我们可以将一些公共的代码抽离成为一个单独的 js 文件，作为一个模块。模块只有通过 module.exports 或者 exports 才能对外暴露接口。

利用模块化可以封装一些工具类，比如，与时间处理的工具类，可以封装在一个模块中；甚至可以为了方便管理某些操作，将所有的操作都都封装到一个模块中.

```
function Logs(logs) {
  console.log(logs)
}

module.exports = {
  Logs: Logs
}
```