实验编号： 13 **四川师大《IOS》实验报告 2018** 年 **12** 月 **12** 日

### **计算机科学学院** 2016 级 4 班 实验名称： 多线程和网络程序设计 \_

姓名：\_谭靖薇\_ 学号：\_2016110437\_ 指导老师：\_\_李贵洋\_\_ 实验成绩:\_\_\_\_\_

**实验 十三 \_**多线程和网络程序设计**\_\_\_\_\_**

1. 实验目的及要求
2. 理解并掌握iOS多线程编程的相关技术；
3. 掌握GCD关键技术，包括block、dispatch等；
4. 掌握WebView的使用；
5. 掌握URLSession的使用，
6. 掌握第三网络库Alamofire的使用方法；
7. 掌握Json的解析。
8. 实验要求
9. 认真填写实验报告，要求附加部分运行界面和主要代码；
10. 对设计好的程序，检查输出是否符合预期，如有错请分析错误原因并解决；
11. 实验内容
12. 采用多线程技术，实现一个大数加程序。
    1. 正确理解DispatchQueue的使用
    2. 从1 到 9999999
    3. 不能阻塞UI主线程
13. Web浏览器;
    1. 使用WebView控件写成一个简易的浏览器，有浏览器的基本功能；
14. 使用网络库进行天气Json数据的解析
    1. APP有两个界面，第一个界面：tableview显示一个城市列表
    2. 第二个界面，显示选择城市的天气数据
    3. 使用第三方网络Alamofire进行网络的连接，获取网络天气数据；
    4. 对获取到的网络数据进行Json的解析；
    5. 天气数据库位置：http://t.weather.sojson.com/api/weather/city/101270101

可选其他 Web API 进行解析:

1. 免费 JSON API: <http://www.sojson.com/api/>
2. 源代码管理 GitHub API: <https://developer.github.com/v3/>
3. 图形识别: <https://imagga.com/>

4. 摄影社区: <https://500px.com/>

1. 实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）
2. 采用多线程技术，实现一个大数加程序。
   1. 正确理解DispatchQueue的使用
   2. 从1 到 9999999
   3. 不能阻塞UI主线程

* 程序代码：

*//*

*// ViewController.swift*

*// MultiThread*

*//*

*// Created by student on 2018/12/15.*

*// Copyright © 2018年 2016110437. All rights reserved.*

*//*

**import** UIKit

**class** ViewController: UIViewController {

**@IBOutlet** **weak** **var** sumlabel: UILabel!

**@IBOutlet** **weak** **var** countLabel: UILabel!

**var** count = 0

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

Timer.scheduledTimer(withTimeInterval: 1, repeats: **true**) { (timer) **in**

**self**.count += 1

print("timer thread:\(Thread.current)")

**self**.countLabel.text = "count:\(**self**.count)"

}

}

**@IBAction** **func** sum(**\_** sender: **Any**) {

**var** sum = 0

DispatchQueue.global().async {

print("sum thread:\(Thread.current)")

**for** i **in** 1...999999999{

sum+=i

DispatchQueue.main.async {

**self**.sumlabel.text = "\(sum)"

}

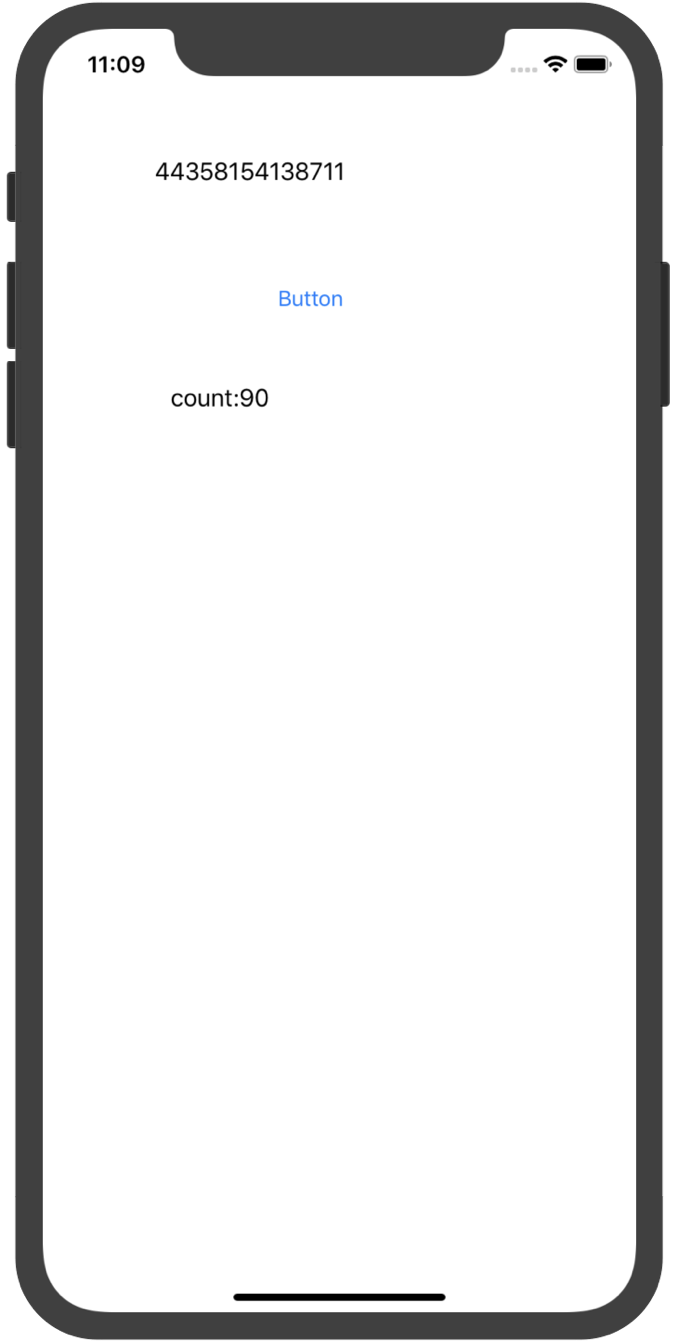
}

}

}

}

* 运行结果：



1. Web浏览器;
   1. 使用WebView控件写成一个简易的浏览器，有浏览器的基本功能；

* 程序代码：

**import** UIKit

**import** WebKit

**class** ViewController: UIViewController,WKNavigationDelegate,WKUIDelegate {

**@IBOutlet** **weak** **var** tfUrl: UITextField!

**@IBOutlet** **weak** **var** webView: WKWebView!

**@IBOutlet** **weak** **var** progressView: UIProgressView!

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

*// Do any additional setup after loading the view, typically from a nib.*

**if** **let** url = URL(string: "http://www.baidu.com") {

webView.load(URLRequest(url: url))

webView.navigationDelegate = **self**

webView.uiDelegate = **self**

webView.addObserver(**self**, forKeyPath: "estimatedProgress", options: .new, context: **nil**)

}

}

**func** webView(**\_** webView: WKWebView, didFinish navigation: WKNavigation!) {

webView.evaluateJavaScript("document.title") { (title, error) **in**

**if** **let** title = title **as**? String {

**self**.title = title

}

}

webView.evaluateJavaScript("ocToJs('test','i hate you')", completionHandler: {(response, error) **in**

print("response:\((response **as**? String) ?? "") error:\(error?.localizedDescription ?? "")")

})

webView.evaluateJavaScript("showAlert()", completionHandler: {(response, error) **in**

print("response:\((response **as**? String) ?? "") error:\(error?.localizedDescription ?? "")")

})

print("Finished!")

}

**@IBAction** **func** go(**\_** sender: **Any**) {

**if** **let** url = URL(string: tfUrl.text ?? "") {

webView.load(URLRequest(url: url))

}

}

**@IBAction** **func** back(**\_** sender: **Any**) {

webView.goBack()

}

**@IBAction** **func** forward(**\_** sender: **Any**) {

webView.goForward()

}

**@IBAction** **func** reload(**\_** sender: **Any**) {

webView.reload()

}

**@IBAction** **func** local(**\_** sender: **Any**) {

**if** **let** url = Bundle.main.url(forResource: "default", withExtension: "html") {

*// webView.load(URLRequest(url: url))*

webView.loadFileURL(url, allowingReadAccessTo: url)

}

}

**func** webView(**\_** webView: WKWebView, runJavaScriptAlertPanelWithMessage message: String, initiatedByFrame frame: WKFrameInfo, completionHandler: **@escaping** () -> Void) {

**let** alert = UIAlertController(title: "Alert", message: message, preferredStyle: UIAlertController.Style.alert)

alert.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: { action **in**

completionHandler()

}))

present(alert, animated: **true**, completion: **nil**)

}

**func** webView(**\_** webView: WKWebView, runJavaScriptConfirmPanelWithMessage message: String, initiatedByFrame frame: WKFrameInfo, completionHandler: **@escaping** (Bool) -> Void) {

**let** alert = UIAlertController(title: "Alert", message: message, preferredStyle: UIAlertController.Style.alert)

alert.addAction(UIAlertAction(title: "Cancel", style: .cancel, handler: { (action) **in**

completionHandler(**false**)

}))

alert.addAction(UIAlertAction(title: "Ok", style: .default, handler: { (action) **in**

completionHandler(**true**)

}))

present(alert, animated: **true**, completion: **nil**)

}

**func** webView(**\_** webView: WKWebView, runJavaScriptTextInputPanelWithPrompt prompt: String, defaultText: String?, initiatedByFrame frame: WKFrameInfo, completionHandler: **@escaping** (String?) -> Void) {

**let** alert = UIAlertController(title: prompt, message: **nil**, preferredStyle: UIAlertController.Style.alert)

alert.addTextField { (textField) **in**

textField.text = defaultText

}

alert.addAction(UIAlertAction(title: "Ok", style: .default, handler: { (action) **in**

completionHandler(alert.textFields?.first?.text ?? "nil")

}))

present(alert, animated: **true**, completion: **nil**)

}

**override** **func** observeValue(forKeyPath keyPath: String?, of object: **Any**?, change: [NSKeyValueChangeKey : **Any**]?, context: UnsafeMutableRawPointer?) {

**if** keyPath == "estimatedProgress" {

progressView.alpha = 1

progressView.progress = Float(webView.estimatedProgress)

**if** progressView.progress == 1 {

progressView.progress = 0

progressView.alpha = 0

}

}

}

**deinit** {

webView.removeObserver(**self**, forKeyPath: "estimatedProgress")

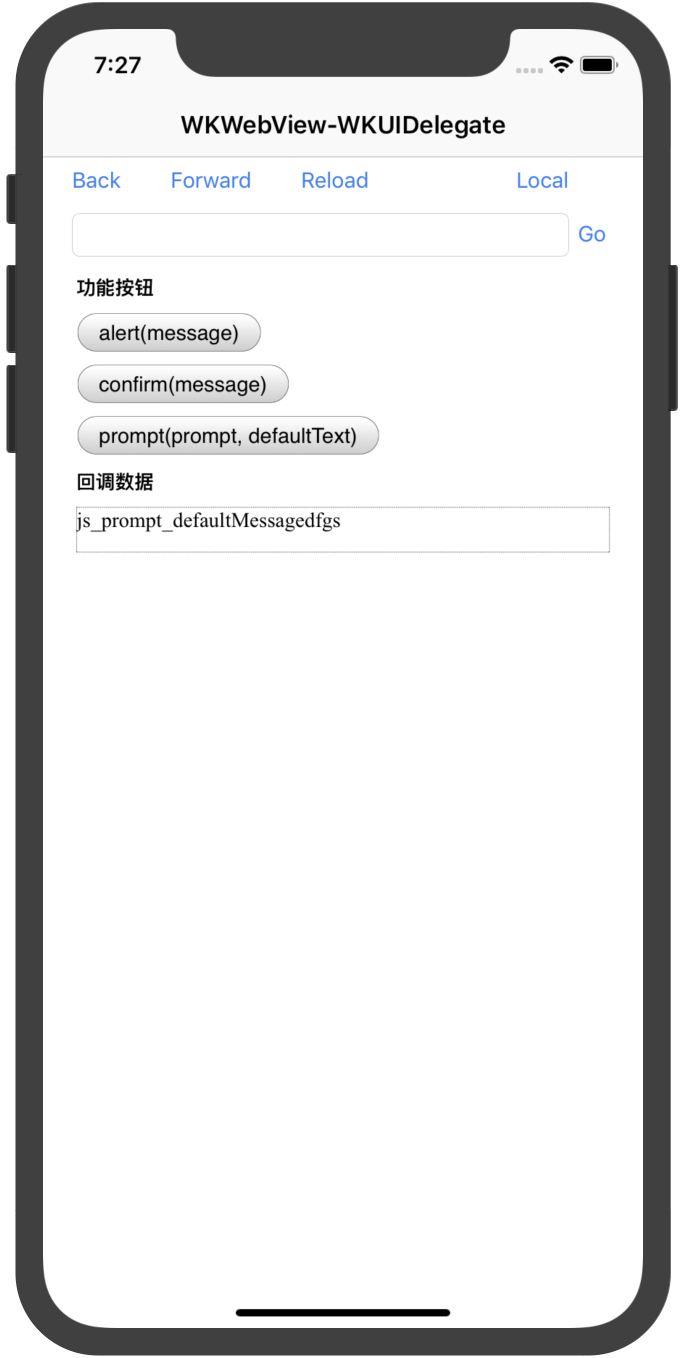
}

}

* 运行结果：





****

1. 使用网络库进行天气Json数据的解析
   1. APP有两个界面，第一个界面：tableview显示一个城市列表
   2. 第二个界面，显示选择城市的天气数据
   3. 使用第三方网络Alamofire进行网络的连接，获取网络天气数据；
   4. 对获取到的网络数据进行Json的解析；
   5. 天气数据库位置：http://t.weather.sojson.com/api/weather/city/101270101

* 程序代码：

JSONViewController

**import** UIKit

**import** Alamofire

**class** JSONViewController: UIViewController {

**let** url = URL(string: "http://t.weather.sojson.com/api/weather/city/101270101")!

**var** weather:AnyObject!

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

*// Do any additional setup after loading the view.*

}

**@IBAction** **func** loadWithSession(**\_** sender: **Any**) {

**let** task = URLSession.shared.dataTask(with: url) { (data, response, err) **in**

**if** **let** json = **try**? JSONSerialization.jsonObject(with: data!, options: .allowFragments) **as** AnyObject{

**self**.weather = json

*// let cityInfo:AnyObject = json.object(forKey: "cityInfo") as AnyObject*

*// let city:NSString = cityInfo.object(forKey: "city") as! NSString*

DispatchQueue.main.async {

**self**.performSegue(withIdentifier: "ShowCity", sender: **self**)

}

}

}

task.resume()

}

**@IBAction** **func** loadWithAF(**\_** sender: **Any**) {

AF.request(url).responseJSON { (response) **in**

**self**.weather = response.value **as** AnyObject

**self**.performSegue(withIdentifier: "ShowCity", sender: **self**)

}

}

*// MARK: - Navigation*

*// In a storyboard-based application, you will often want to do a little preparation before navigation*

**override** **func** prepare(for segue: UIStoryboardSegue, sender: **Any**?) {

*// Get the new view controller using segue.destination.*

*// Pass the selected object to the new view controller.*

**if** segue.identifier == "ShowCity" {

**if** **let** secVC = segue.destination **as**? CityTableViewController {

secVC.weather = **self**.weather

}

}

}

}

CityTableViewController

**import** UIKit

**import** Alamofire

**class** CityTableViewController: UITableViewController {

**var** weather:AnyObject!

**var** infoData:AnyObject!

**let** url = URL(string: "http://10.0.1.2/piclist.php")!

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

infoData = (weather.object(forKey: "data") **as** AnyObject)

*// Uncomment the following line to preserve selection between presentations*

*// self.clearsSelectionOnViewWillAppear = false*

*// Uncomment the following line to display an Edit button in the navigation bar for this view controller.*

*// self.navigationItem.rightBarButtonItem = self.editButtonItem*

}

**@IBAction** **func** reload(**\_** sender: **Any**) {

**self**.refreshControl?.beginRefreshing()

AF.request(url).responseJSON { (response) **in**

**self**.weather = response.value **as** AnyObject

**self**.refreshControl?.endRefreshing()

**self**.tableView.reloadData()

}

}

*// MARK: - Table view data source*

**override** **func** numberOfSections(in tableView: UITableView) -> Int {

*// #warning Incomplete implementation, return the number of sections*

**return** 1

}

**override** **func** tableView(**\_** tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

*// #warning Incomplete implementation, return the number of rows*

**return** (weather?.count)!-1 ?? 0

}

**override** **func** tableView(**\_** tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

**let** cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)

*// Configure the cell...*

**let** d = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "date") **as**? String

*// cell.textLabel?.text = (weather.object(forKey: "cityInfo") as AnyObject).object(forKey: "city") as? String*

**let** c = (weather.object(forKey: "cityInfo") **as** AnyObject).object(forKey: "city") **as**? String

**let** all = c!+d!

cell.textLabel?.text = all

**return** cell

}

*/\**

*// Override to support conditional editing of the table view.*

*override func tableView(\_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {*

*// Return false if you do not want the specified item to be editable.*

*return true*

*}*

*\*/*

*/\**

*// Override to support editing the table view.*

*override func tableView(\_ tableView: UITableView, commit editingStyle: UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {*

*if editingStyle == .delete {*

*// Delete the row from the data source*

*tableView.deleteRows(at: [indexPath], with: .fade)*

*} else if editingStyle == .insert {*

*// Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view*

*}*

*}*

*\*/*

*/\**

*// Override to support rearranging the table view.*

*override func tableView(\_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to: IndexPath) {*

*}*

*\*/*

*/\**

*// Override to support conditional rearranging of the table view.*

*override func tableView(\_ tableView: UITableView, canMoveRowAt indexPath: IndexPath) -> Bool {*

*// Return false if you do not want the item to be re-orderable.*

*return true*

*}*

*\*/*

*// MARK: - Navigation*

*// In a storyboard-based application, you will often want to do a little preparation before navigation*

**override** **func** prepare(for segue: UIStoryboardSegue, sender: **Any**?) {

*// Get the new view controller using segue.destination.*

*// Pass the selected object to the new view controller.*

**if** segue.identifier == "ShowDetail" {

**if** **let** secVC = segue.destination **as**? DetailViewController {

**if** **let** indexPath = tableView.indexPath(for: sender **as**! UITableViewCell) {

**var** date = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "date") **as**? String

**var** sunrise = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "sunrise") **as**? String

**var** high = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "high") **as**? String

**var** low = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "low") **as**? String

**var** sunset = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "sunset") **as**? String

**var** aqi = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "aqi") **as**? String

**var** ymd = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "ymd") **as**? String

**var** week = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "week") **as**? String

**var** fx = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "fx") **as**? String

**var** fl = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "fl") **as**? String

**var** type = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "type") **as**? String

**var** notice = (((infoData.object(forKey: "forecast")**as**! NSArray)[indexPath.row])**as** AnyObject).object(forKey: "notice") **as**? String

**var** wea = Array<String>()

wea.append(date!)

wea.append(week!)

wea.append(sunrise!)

wea.append(sunset!)

wea.append(high!)

wea.append(low!)

wea.append(ymd!)

wea.append(ymd!)

wea.append(fx!)

wea.append(fl!)

wea.append(type!)

wea.append(notice!)

*// [date,week,sunrise,sunset,high,low,aqi,ymd,fx,fl,type,notice]*

secVC.infoData = wea **as**! [String]

}

}

}

}

}

DetailViewController

**import** UIKit

**import** Alamofire

**class** DetailViewController: UIViewController {

**@IBOutlet** **weak** **var** today: UILabel!

**@IBOutlet** **weak** **var** week: UILabel!

**@IBOutlet** **weak** **var** sunrise: UILabel!

**@IBOutlet** **weak** **var** sunset: UILabel!

**@IBOutlet** **weak** **var** high: UILabel!

**@IBOutlet** **weak** **var** low: UILabel!

**@IBOutlet** **weak** **var** aci: UILabel!

**@IBOutlet** **weak** **var** ymd: UILabel!

**@IBOutlet** **weak** **var** fx: UILabel!

**@IBOutlet** **weak** **var** fl: UILabel!

**@IBOutlet** **weak** **var** type: UILabel!

**@IBOutlet** **weak** **var** notice: UILabel!

*//*

*// var infoData:[String]?*

**var** infoData = Array<String>()

**override** **func** viewDidLoad() {

**super**.viewDidLoad()

today.text = "时间："+(infoData[0])

week.text = "星期："+(infoData[1])

sunrise.text = "日出："+(infoData[2])

sunset.text = "日落："+(infoData[3])

high.text = "最高温度："+(infoData[4])

low.text = "最低温度："+(infoData[5])

aci.text = "aci："+(infoData[6])

ymd.text = "ymd："+(infoData[7])

fx.text = "fx："+(infoData[8])

fl.text = "fl："+(infoData[9])

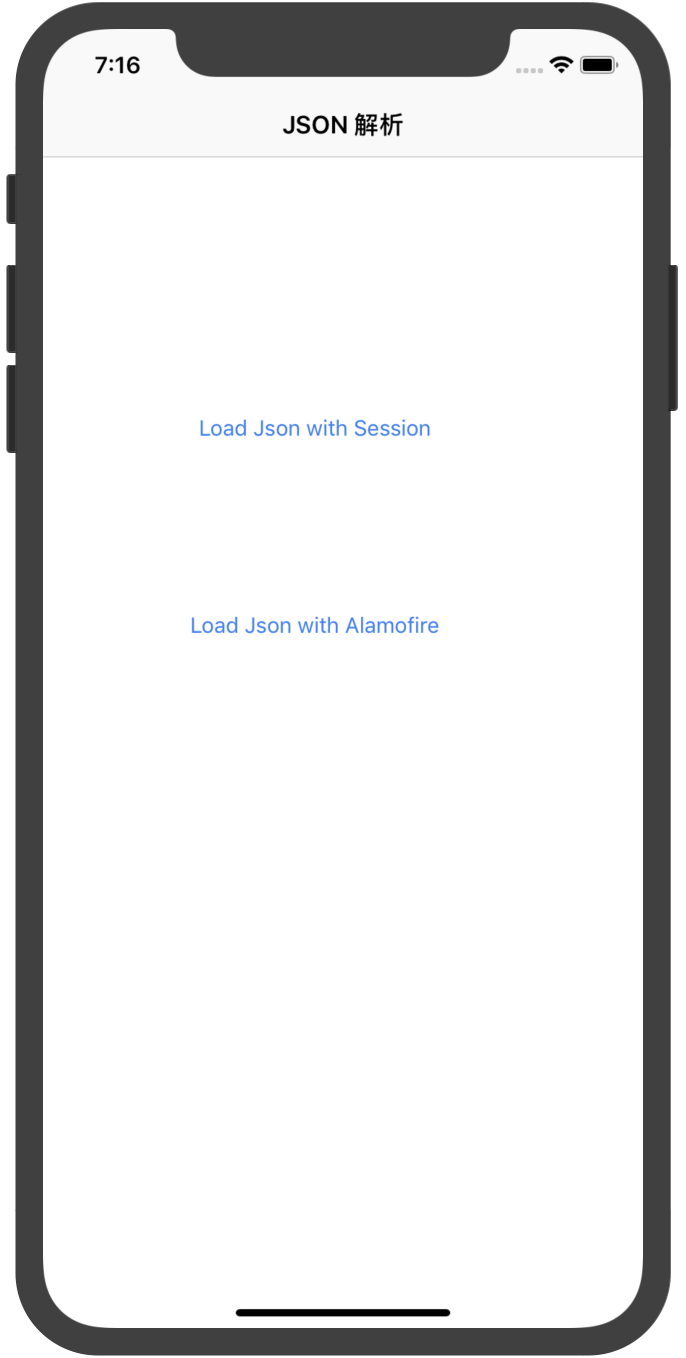
type.text = "类型："+(infoData[10])

notice.text = "注意事项："+(infoData[11])

}

}

* 运行结果：







可选其他 Web API 进行解析:

1. 免费 JSON API: <http://www.sojson.com/api/>
2. 源代码管理 GitHub API: <https://developer.github.com/v3/>
3. 图形识别: <https://imagga.com/>

4. 摄影社区: <https://500px.com/>

1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

Github地址：

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。