luaLib接入

luaLib接入指南

接入步骤

1.执行pod init

2.在Podfile的target中，加入如下代码。install后记得重新打开工程

﻿

project\_name = `ls | grep .xcodeproj`.strip

project = Xcodeproj::Project.open(project\_name)

def changeCfg (config, attr)

if config.build\_settings[attr].is\_a? String then

config.build\_settings[attr] = [config.build\_settings[attr]]

end

if config.build\_settings[attr] and not config.build\_settings[attr].include? "$(inherited)" then

config.build\_settings[attr] << "$(inherited)"

puts "[" + config.name + "] auto add \"$(inherited)\":" + attr

end

end

project.targets.each do |target|

target.build\_configurations.each do |config|

changeCfg(config, "OTHER\_LDFLAGS")

changeCfg(config, "LIBRARY\_SEARCH\_PATHS")

changeCfg(config, "HEADER\_SEARCH\_PATHS")

changeCfg(config, "GCC\_PREPROCESSOR\_DEFINITIONS")

@INFOPLIST = config.build\_settings["INFOPLIST\_FILE"]

`/usr/libexec/PlistBuddy -c "Delete :NSAppTransportSecurity" "#{@INFOPLIST}"`

`/usr/libexec/PlistBuddy -c "Add :NSAppTransportSecurity" "#{@INFOPLIST}"`

`/usr/libexec/PlistBuddy -c "Add :NSAppTransportSecurity:NSAllowsArbitraryLoads bool true" "#{@INFOPLIST}"`

end

end

project.save

pod 'lualibPod', :git => 'https://。。。。。。/lualibPod.git'

﻿

3.在main.m 处调用

修改main.m 将原来工程正常运行的代码段放到if条件成立的代码块中。

﻿

#include <lualib/luacore.h>

int main(int argc, char \*argv[]) {

// 1. 小游戏模式的代码：

initLuaCore();

// 然后是小游戏的入口.....

​

// -------------分割线---------------------

​

// 2. 外部模式的代码：

if(initLuaCore()){

// isOpen = true 的入口

} else {

// isOpen = false 入口

};

​

// -------------分割线---------------------

​

// 3. 其他地方获得开关的方法

short isOpen = getSystemTag();

}

﻿

5.如果链接报错。可以删除xcconfig中-objC的标记

接入参数设置

工程内部使用json文件加密到图片中进行配置。图片的名称必须是：LaunchScreenBackground.png

Json支持的字段如下：

url string|array 如果是一个配置。则配置成字符串。如果是多个。就配置成字符串数组。如果只用时间控制的话，就不需要配置。

time string 设置时间 格式为 "20190101"

io boolean 是否是对外模式。不对外的时候，isOpen则会进入到内部h5网页，对外则只返回结果

远程参数设置

远程配置使用json配置接入

- isOpen boolean 表示是否跳转，默认false

- url string 跳转的链接，默认为空

- o number 1 竖屏 2 横屏 其他数字均为自动旋转， 默认为自动旋转

- timeZ array 允许的时区列表（比如东8区，则是3600\*8=28800 则为：[28800]）。默认为[]。当系统时区不包含在设定的时区内时，会进入到审核模式。如果自定义此列，则会覆盖配置。配置为空数组时。则不受时区限制。

- lang string 允许进入的系统语言（比如中文为zh，英文为en)，默认为空。存在此列时，会进行系统语言判定。系统语言如果不等于设定的语言时，会进入到审核模式。如果自定义此列，则会覆盖默认配置。如果不想开启这个开关。可以设置这一列为空字符串。

- patch array 需要用默认浏览器打开的链接列表(匹配前缀)

- full boolean 是否在全面屏中强制全屏，默认false

- ignoreMenuGap boolean 是否在全面屏中竖屏中忽略菜单栏间距。默认false

- hideNav boolean 是否隐藏下方导航条。默认false

- goh5 boolean 在外部开关模式下。强制跳到h5

- noh5 boolean 在非外部开关模式下。强制跳到外部

- set boolean 网络不好时。是否显示设置按钮，默认不显示。

- version string 用于设置当前可用的最大版本号。高于这个版本号的默认为审核模式。低于或等于这个版本号才会使用此配置。

- noload boolean 是否关闭加载功能。默认false

- close boolean 是否关闭当前的配置方式。默认false。用于解决多个配置方案同时可以生效时，此开关可以手动关闭没必要的配置，防止冲突。

idfa、idfv等参数接入说明

sdk提供使用get参数占位符的方式进行传递参数。其中

idfa的占位符是 \_\_IDFA\_\_

idfv的占位符是 \_\_IDFV\_\_