|  |  |
| --- | --- |
| NAVER Cafe SDK的使用说明 | |
|  |  |

|  |
| --- |
| 著作权 |

Copyright © NAVER Corp. All Rights Reserved.

该文件是NAVER（株）的知识资产，因此在任何条件下没有NAVER（株）的正式批准，不得擅自复印、传送、发行或是更改使用该文件的一部分或是全部内容。

该文件仅用于提供信息。尽管NAVER（株）为了检测该文件内信息的完整性和真实性竭尽努力，但是对于内容中可能存在的错误和遗漏并不负责。因此伴随着该文件的使用或是使用结果所产生的责任全部归用户所有，NAVER（株）对此不进行明确或是隐含的任何保证。

包含相关的URL信息，在该文件中提到的特定软件商品或是产品依照相应持有者所属当地和国内外相关法律，用户自身全权负责由于不遵守相应法律而导致的所有结果。

NAVER（株）可以不事先预告而更改该文件的内容。

|  |
| --- |
| 文件信息 |

文件概要

该文件是说明NAVER Cafe SDK适用于应用程序的方法。

读者

该文件的读者是想将NAVER Cafe SDK用于应用程序的开发者。

咨询处

如该文件内容有误或是有与内容相关的疑问，可以用下面的联系方式进行咨询。

联系方式: [dl\_gamesdkpartner@navercorp.com](mailto:dl_gamesdkpartner@navercorp.com)

文件版本及介绍

|  |  |
| --- | --- |
| 日期 | 介绍 |
| 2015/12/23 | 最终发行 |

|  |
| --- |
| 标记规则 |

参考标记

参考

记述读者应当参考的内容。

注意标记

注意

记述读者必须要知道的事项、可能引发系统错误的事项。

UI 语句，用户输入值标记

* UI 语句： 菜单 > 下级菜单
* 用户固定输入值：输入localhost。
* 用户可变输入值：<http://www.naver.com/>{公司名}

源代码标记

该文件中源代码用灰底色、黑字体标记。

COPYDATASTRUCT st;

st.dwData = PURPLE\_OUTBOUND\_ENDING;

st.cbData = sizeof(pp);

st.lpData = &pp;

::SendMes(GetTargetHwnd(), WM\_COPYDATA, (WPARAM)this->m\_hWnd, (LPARAM)&st);

|  |
| --- |
| 目录 |

[概要 9](#_Toc438643801)

[NAVER Cafe SDK概要 9](#_Toc438643802)

[NAVER Cafe SDK功能及特点 9](#_Toc438643803)

[主要功能 9](#_Toc438643804)

[使用环境 9](#_Toc438643805)

[用于Android应用程序的说明 11](#_Toc438643806)

[开发环境 11](#_Toc438643807)

[软件要求事项 11](#_Toc438643808)

[资源库构成 11](#_Toc438643809)

[设置开发环境 12](#_Toc438643810)

[使用NAVER Cafe SDK 12](#_Toc438643811)

[设置AndroidManifest.xml 12](#_Toc438643812)

[NAVER Cafe SDK初始化 12](#_Toc438643813)

[开启NAVER Cafe SDK画面 13](#_Toc438643814)

[关闭NAVER Cafe SDK画面 13](#_Toc438643815)

[写文章 13](#_Toc438643816)

[处理App Scheme 13](#_Toc438643817)

[映射用户游戏ID 14](#_Toc438643818)

[更改资源图像 14](#_Toc438643819)

[API 参照 14](#_Toc438643820)

[Glink 14](#_Toc438643821)

[init() 14](#_Toc438643822)

[isShowGlink() 15](#_Toc438643823)

[popBackStack() 15](#_Toc438643824)

[setGameUserId() 16](#_Toc438643825)

[startEvent() 16](#_Toc438643826)

[startImageWrite() 17](#_Toc438643827)

[startHome() 17](#_Toc438643828)

[startMenu() 18](#_Toc438643829)

[startNotice() 18](#_Toc438643830)

[startProfile() 19](#_Toc438643831)

[startVideoWrite() 19](#_Toc438643832)

[startWrite() 20](#_Toc438643833)

[stop() 20](#_Toc438643834)

[用于 iOS应用程序的说明 21](#_Toc438643835)

[开发环境 21](#_Toc438643836)

[软件要求事项 21](#_Toc438643837)

[资源库构成 21](#_Toc438643838)

[设置开发环境 21](#_Toc438643839)

[使用NAVER Cafe SDK 24](#_Toc438643840)

[NAVER Cafe SDK初始化 24](#_Toc438643841)

[设置 NAVER Login 24](#_Toc438643842)

[开启NAVER Cafe SDK 画面 25](#_Toc438643843)

[写文章 25](#_Toc438643844)

[API 参照 25](#_Toc438643845)

[NCSDKManager 25](#_Toc438643846)

[@property (nonatomic, weak) id parentViewController 25](#_Toc438643847)

[(void)dismissViewController 26](#_Toc438643848)

[(void)dismissTopViewController 26](#_Toc438643849)

[(NCSDKManager \*)getSharedInstance 27](#_Toc438643850)

[(id)navercafeRootViewController 27](#_Toc438643851)

[(void)resetSharedInstance 27](#_Toc438643852)

[(void)presentArticlePostViewControllerWithMenuId 28](#_Toc438643853)

[(void)presentArticlePostViewControllerWithType 28](#_Toc438643854)

[(void)presentMainViewController 29](#_Toc438643855)

[(void)presentMainViewControllerWithArticleId 29](#_Toc438643856)

[(void)presentMainViewControllerWithTabIndex 30](#_Toc438643857)

[(void)presentViewController 30](#_Toc438643858)

[(void)setGameUserId 31](#_Toc438643859)

[(void)setNaverLoginClientId 31](#_Toc438643860)

[NCSDKLoginManager 31](#_Toc438643861)

[@property (nonatomic, weak) UIViewController \*rootViewController 32](#_Toc438643862)

[(NSString \*)accessToken 32](#_Toc438643863)

[(NSString \*)accessTokenExpireTime 32](#_Toc438643864)

[(BOOL)finishNaverLoginWithURL 33](#_Toc438643865)

[(NCSDKLoginManager \*)getSharedInstance 33](#_Toc438643866)

[(void)isLoginWithFinish 34](#_Toc438643867)

[(BOOL)isValidAccessTokenExpireTimeNow 34](#_Toc438643868)

[(void)loginWithFinish 34](#_Toc438643869)

[(void)logout 35](#_Toc438643870)

[(void)refreshAccessToken 35](#_Toc438643871)

[(void)refreshAccessTokenWithFinish 36](#_Toc438643872)

[(void)requestDeleteToken 36](#_Toc438643873)

[(void)setIsInAppOauthEnable 36](#_Toc438643874)

[(void)setIsNaverAppOauthEnable 37](#_Toc438643875)

[用于Unity 应用程序的说明 38](#_Toc438643876)

[开发环境 38](#_Toc438643877)

[软件要求事项 38](#_Toc438643878)

[资源库构成 38](#_Toc438643879)

[设置开发环境 38](#_Toc438643880)

[使用NAVER Cafe SDK 42](#_Toc438643881)

[运行 42](#_Toc438643882)

[映射用户的游戏ID和 NAVER ID 42](#_Toc438643883)

[替换AFNetworking 42](#_Toc438643884)

[API 参照 43](#_Toc438643885)

[GLinkUnity 43](#_Toc438643886)

[executeArticlePost() 43](#_Toc438643887)

[executeArticlePostWithImage() 43](#_Toc438643888)

[executeArticlePostWithVideo() 44](#_Toc438643889)

[executeMain() 44](#_Toc438643890)

|  |
| --- |
| 图表和图片目录 |

图表目录

[图 1 NAVER Cafe SDK 9](#_Toc438643674)

图片目录

[表 1 使用Android用NAVER Cafe SDK时必需的资源库 11](#_Toc438546478)

[表 2 使用iOS用NAVER Cafe SDK必需的资源库 21](#_Toc438546479)

# 概要

## NAVER Cafe SDK概要

NAVER Cafe SDK是与游戏引擎无关，在手游中可以简捷统合NAVER Cafe的资源库。使用NAVER Cafe SDK时，游戏玩家不离开游戏就可与游戏社区（NAVER Cafe）进行沟通。

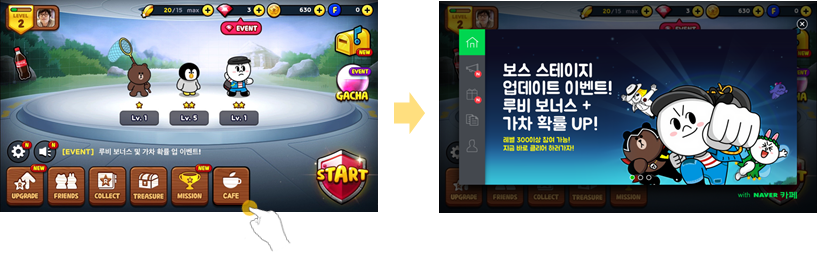


图 1 NAVER Cafe SDK

## NAVER Cafe SDK功能及特点

### 主要功能

NAVER Cafe SDK的主要功能如下

* 支持iOS和Android
* 支持Unity 4以上及Cocos2d-x 2.1以上游戏引擎
* 提供可以使用Cafe功能的画面
* 认证使用OAuth 2.0版本的NAVER ID进行登录的用户

### 使用环境

#### 操作系统

NAVER Cafe SDK在如下环境中运作。

* Android: Android 4.2（Jelly Bean）API Level 17以上
* iOS: iOS 7.0以上（支持ARMv7, ARMv7s, ARM64）

#### 登录NAVER Login

如在NAVER Cafe SDK中通过使用“NAVER Login”功能认证用户时，需有客户端ID和客户端密码。

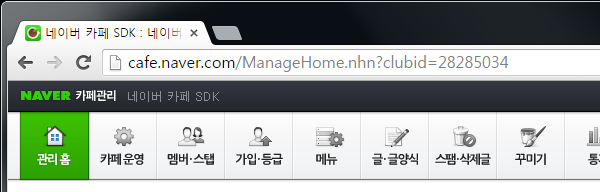
* 客户端ID：在NAVER Login上登录游戏后得到的客户端ID
* 客户端密码：在NAVER Login上登录游戏后得到的客户端密码

在NAVER Login上登录应用程序就可以得到客户端ID和客户端密码。请参考NAVER Login Developers的文件或向负责人咨询([dl\_gamesdkpartner@navercorp.com](mailto:dl_gamesdkpartner@navercorp.com))。

* NAVER Login Developers(韩国语)：<https://nid.naver.com/devcenter/main.nhn>

#### NAVER Cafe ID

如想使用NAVER Cafe SDK，需有适用的NAVER Cafe的Cafe ID。Cafe ID是在管理NAVER Cafe页面的 URL中的clubid参数的值。



# 用于Android应用程序的说明

## 开发环境

### 软件要求事项

#### 开发工具

* IDE: Android Studio或是Eclipse
* Android Support Library v7

#### 资源库

如想使用NAVER Cafe SDK，需将下面的资源库添加到项目中一起建立。

表 1 使用Android用NAVER Cafe SDK时必需的资源库

|  |  |
| --- | --- |
| 资源库 | 下载URL |
| NAVER Login | * 包含在NAVER Cafe SDK资源库文件中（4.1.4 版本）。 * 下载URL: <https://static.nid.naver.com/images/web/devcenter/3rdparty_login_library_android_4.1.4.zip> |
| NAVER Volleyer | * 包含在NAVER Cafe SDK资源库文件中（2.0.1 版本）。 * 下载URL: <http://mvnrepository.com/artifact/com.navercorp.volleyextensions/volleyer> |
| Volley | * 包含在NAVER Cafe SDK资源库文件中（1.0.2 版本）。 * 下载URL: [http://mvnrepository.com/artifact/com.mcxiaoke.volley/library](http://mvnrepository.com/artifact/com.mcxiaoke.volley/library/) |
| Google Gson | * 包含在NAVER Cafe SDK资源库文件中（2.3.1 版本）。 * 下载URL: <http://mvnrepository.com/artifact/com.google.code.gson/gson> |
| Glide | * 包含在NAVER Cafe SDK资源库文件中（3.6.1 版本）。 * 下载URL: <http://mvnrepository.com/artifact/com.github.bumptech.glide/glide> |
| Otto | * 包含在NAVER Cafe SDK资源库文件中(1.3.8 版本）。 * 下载 URL: <http://mvnrepository.com/artifact/com.squareup/otto> |

### 资源库构成

Android用NAVER Cafe SDK资源库如下构成。

* lib: NAVER Cafe SDK资源库和必需资源库
  + cafeSdk-x.x.x.aar: 在Android Studio中使用的NAVER Cafe SDK资源库文件
  + cafeSdk-x.x.x.zip: 在Eclipse中使用的NAVER Cafe SDK资源库
  + library: 使用NAVER Cafe SDK时必需的资源库文件夹
* sample: 使用NAVER Cafe SDK资源库的示例项目文件夹
  + navercafesdk-sample-android-studio: 可以在Android Studio中使用的NAVER Cafe SDK示例项目文件夹
  + navercafesdk-sample-eclipse-master: 可以在Eclipse中使用的NAVER Cafe SDK示例项目文件夹

### 设置开发环境

如想使用NAVER Cafe SDK，根据IDE如下设置开发环境。

#### Android Studio

* + - * 1. 在Android项目的libs文件夹里复制NAVER Cafe SDK资源库文件（cafeSdk-x.x.x.aar）。
        2. 在Android项目的libs文件夹里复制NAVER Login资源库文件（naveroauthlogin-4.x.x.jar）。
        3. 在菜单里点击File > Project Structure，并在Project Structure对话框里点击App > Dependencies后添加资源库。或如下在Android项目的build.gladle文件中直接添加资源库。

compile 'com.android.support:support-v13:23.1.0'

compile 'com.navercorp.volleyextensions:volleyer:2.0.1'

compile 'com.google.code.gson:gson:2.3.1'

compile 'com.github.bumptech.glide:glide:3.6.1'

compile 'com.squareup:otto:1.3.8'

#### Eclipse

* + - * 1. 解除对NAVER Cafe SDK资源库文件（cafeSdk-x.x.x.zip）的压缩。
        2. 在解除了压缩的NAVER Cafe SDK资源库文件夹下面的libs文件夹中添加构建时必需的资源库文件。
        3. 在Android项目的AndroidManifest.xml文件中如下添加在NAVER Login中使用的活动和在NAVER Cafe SDK中使用的活动。

<activity

android:name="com.naver.glink.android.sdk.ui.VideoPlayActivity"

android:screenOrientation="sensorLandscape"

android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen" />

<activity

android:name="com.nhn.android.naverlogin.ui.OAuthLoginActivity"

android:screenOrientation="sensorLandscape"

android:theme="@android:style/Theme.Translucent.NoTitleBar" />

<activity

android:name="com.nhn.android.naverlogin.ui.OAuthLoginInAppBrowserActivity"

android:label="OAuth2.0 In-app"

android:screenOrientation="sensorLandscape" />

## 使用NAVER Cafe SDK

### 设置AndroidManifest.xml

在Android项目的AndroidManifest.xml文件中添加如下内容。

在package属性里输入在NAVER Login登录应用程序时在Android Intent设置的内容。一并设置对调控NAVER Cafe SDK的客体Glink Class的使用权限。

<!—package要与用 NAVER Login Developers 登录的 Android Intent相同。-->

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.naver.glink.sample">

<!—对Glink需要的实用的权限 -->

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

### NAVER Cafe SDK初始化

初始化NAVER Cafe SDK的功能如下用Glink.init()方法实现。

客户端ID和客户端密码是在 NAVER Login 上登录应用程序后得到的值。

/\*\*

\* 用在NAVER Login Developers得到的信息初始化SDK。

\* 调出Glink的其他方法前必须先进行初始化。

\* 开发者中心地址: https://nid.naver.com/devcenter/main.nhn

\*/

final int cafeId = 28266581;

final String consumerKey = "客户端ID";

final String consumerSecret = "客户端密码";

Glink.init(consumerKey, consumerSecret, cafeId);

### 开启NAVER Cafe SDK画面

NAVER Cafe SDK画面由5个标签构成。开启NAVER Cafe SDK时选择以何种选定标签的状态开始。

例如如想以选定主标签页的状态开始，就如下使用Glink.startHome()方法。

// 以主页面开始。

Glink.startHome(activity);

除此之外如想以公告标签页、活动标签页、留言板标签页、个人资料标签页开始，就使用下面的方法。

* Glink.startNotice()方法: 以公告标签页开始。
* Glink.startEvent()方法: 以 活动标签页开始。
* Glink.startMenu()方法: 以 留言板标签页开始。
* Glink.startProfile()方法: 以个人资料标签页开始。

### 关闭NAVER Cafe SDK画面

关闭NAVER Cafe SDK画面的功能用Glink.stop()方法或Glink.popBackStack()方法实现。

* Glink.stop()方法：立即关闭画面并结束NAVER Cafe SDK。
* Glink.popBackStack()方法：用backStack一个个关闭累积的画面。关闭所有画面后结束NAVER Cafe SDK。可以用backStack累积的画面为查看文章画面、写文章画面、检索画面。

下面是用Glink.stop()方法关闭NAVER Cafe SDK画面的例子。

// 用stop()方法关闭。

Glink.stop(activity);

下面是用Glink.popBackStack()方法关闭NAVER Cafe SDK画面的例子。

// 用popBackStack关闭。

Glink.popBackStack(activity);

### 写文章

游戏玩家在Cafe写文章的画面如下用Glink.startWrite()方法实现。

// 填入基本标题，正文，开启写文章画面。

如是int menuId = 4; // 0，就不选择菜单。

String text = " 填入基本标题,正文，开启写文章画面。";

Glink.startWrite(MainActivity.this, "subject", text);

使用Glink.startImageWrite()方法或是Glink.startVideoWrite()方法，就可以运行添加图片或视频的状态下执行留言板写文章画面。

### 处理App Scheme

有用App Scheme移动的广告条图像时如下设置OnClickAppSchemeBannerListener，那么在有触碰活动时就可以实现处理App Scheme的功能。

// AppScheme touch Listener 设置。

Glink.setOnClickAppSchemeBannerListener(new Glink.OnClickAppSchemeBannerListener() {

@Override public void onClickAppSchemeBanner(String appScheme) {

// 在Cafe管理中设置的App Scheme 字符串转交到NAVER Cafe SDK。

// 实现处理App Scheme的代码。

}

});

### 映射用户游戏ID

映射用户的游戏ID和NAVER ID的功能如下用Glink.setGameUserId()方法实现。

// 映射用户的游戏ID和NAVER ID。

// 在个人资料画面可以确认映射的游戏ID。

Glink.setGameUserId(this, "gameUserId", "游戏 ID");

### 更改资源图像

包含在NAVER Cafe SDK资源库中的资源图像可以如下进行更改。

* + - * 1. 解除对NAVER Cafe SDK资源库文件（.aaa文件）的压缩。
        2. 解除压缩的文件夹里的/res/drawable-xhdpi文件夹中的图片可以更改为所期望的图片。
        3. 重新压缩已解除压缩的文件夹成NAVER Cafe SDK资源库文件（.aaa文件）。
        4. 使用新压缩的NAVER Cafe SDK资源库构建项目。

注意

更改图片时图片的大小必须与现有图片相同。

## API 参照

### Glink

调控NAVER Cafe SDK的Class。使用Glink Class的方法实现初始化或是开启、结束NAVER Cafe SDK的功能。

Glink Class的方法如下。

* init()
* isShowGlink()
* popBackStack()
* setGameUserId()
* startEvent()
* startImageWrite()
* startHome()
* startMenu()
* startNotice()
* startProfile()
* startVideoWrite()
* startWrite()
* stop()

### init()

#### 说明

初始化NAVER Cafe SDK。

#### 文章版面

public static void init(String clientId, String clientSecret, int cafeId);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| clientId | String | Y | 客户端ID。在NAVER Login上登录应用程序后得到的值。 |
| clientSecret | String | Y | 客户端密码。在NAVER Login上登录应用程序后得到的值。 |
| cafeId | int | Y | Cafe ID。在NAVER Cafe 管理页面的URL中clubid 参数的值。 |

#### 返回值

无

#### 代码例子

// 使用用NAVER ID的登录信息和Cafe ID初始化NAVER Cafe SDK。

Glink.init(“abcd”, "aaaa", 33);

### isShowGlink()

#### 说明

确认NAVER Cafe SDK画面是否打开着。

#### 文章版面

public static boolean isShowGlink(Activity activity);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

* true: 打开NAVER Cafe SDK画面的状态
* false: 关闭NAVER Cafe SDK画面的状态

代码例子

// 确认NAVER Cafe SDK画面是否打开着。

// 如返回值为true，即为打开状态,如返回值为false即为关闭状态。

Glink.isShowGlink(this);

### popBackStack()

#### 说明

用backStack一个个关闭积累的画面。关闭所有画面后，就结束NAVER Cafe SDK。可以用backStack积累的画面为 查看文章画面、写文章画面、检索画面。

#### 文章版面

public static void popBackStack(Activity activity);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

无

代码例子

// 一个个关闭NAVER Cafe SDK画面。

Glink.popBackStack (activity);

### setGameUserId()

#### 说明

映射用户的游戏ID和Cafe ID。

#### 文章版面

public static void setGameUserId(Activity activity, String gameUserId, String fieldName);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |
| gameUserId | String | N | 用户的游戏ID |
| fieldName | String | N | 要在个人资料画面标识的ID（基本值：用户的游戏ID） |

#### 返回值

无

#### 代码例子

// 映射用户的游戏ID和NAVER ID。

// 可以在个人资料画面确认被映射的游戏ID。

Glink.setGameUserId(this, "gameUserId", "游戏ID");

### startEvent()

#### 说明

以选定活动标签页的状态开启NAVER Cafe SDK。

#### 文章版面

public static void startEvent(Activity activity);

#### 参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

无

代码例子

// 以活动标签页开启NAVER Cafe SDK。

Glink.startEvent(activity);

### startImageWrite()

#### 说明

打开附加了图片的留言板写文章画面。

#### 文章版面

public static void startImageWrite(Activity activity, int menuId, String subject, String text, String imagery);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |
| menuId | int | N | 留言板ID（基本值：-1）。留言板ID是NAVER Cafe留言板URL中menuid参数的值。 |
| subject | String | N | 文章标题 |
| text | String | N | 文章正文 |
| imageUri | String | N | 图片文件的径路（URI 形式） |

返回值

无

代码例子

// 有基本标题，正文，以及添加图片的情况下，开启写文章画面。图片以URI形式填入。

如为int menuId = 4; // 0，就不能选择菜单。

String text = " 有基本标题，正文，以及添加图片的情况下，开启写文章画面。图片以URI形式填入。";

String path = "your image uri";

Glink.startImageWrite(MainActivity.this, menuId, "subject", text, path);

### startHome()

#### 说明

以选定主标签页的状态打开NAVER Cafe SDK画面。

#### 文章版面

public static void startHome(Activity activity);

#### 参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

#### 返回值

无

#### 代码例子

// 以主标签页打开NAVER Cafe SDK 画面。

Glink.startHome(activity);

### startMenu()

#### 说明

以选定留言板标签页的状态打开NAVER Cafe SDK画面。

#### 文章版面

public static void startMenu(Activity activity);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

无

代码例子

// 以留言板标签页打开NAVER Cafe SDK画面。

Glink.startMenu(activity);

### startNotice()

#### 说明

以选定公告标签页的状态打开NAVER Cafe SDK 画面。

#### 文章版面

public static void startNotice(Activity activity);

#### 参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

无

代码例子

// 以公告标签页打开NAVER Cafe SDK画面。

Glink.startNotice(activity);

### startProfile()

#### 说明

以选定个人资料标签页的状态打开NAVER Cafe SDK 画面。

#### 文章版面

public static void startProfile(Activity activity);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

无

代码例子

// 用个人资料标签页打开NAVER Cafe SDK画面。

Glink.startProfile(activity);

### startVideoWrite()

#### 说明

以附加视频的状态打开留言板写文章画面。

#### 文章版面

public static void startVideoWrite(Activity activity, int menuId, String subject, String text, String videoUri);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |
| menuId | int | N | 留言板ID（基本值：-1）。留言板ID是NAVER Cafe留言板UR中menuid参数的值。 |
| subject | String | N | 文章标题 |
| text | String | N | 文章正文 |
| videoUri | String | N | 视频文件的径路（URI形式） |

返回值

无

代码例子

// 有基本标题，正文，以及添加录像的情况下，开启写文章画面。 图片以URI形式填入。

如是int menuId = 4; // 0，则不能选择菜单。

String text = " 有基本标题，正文，以及添加录像的情况下，开启写文章画面。 图片以URI形式填入。";

String path = "your video uri";

Glink.startVideoWrite(MainActivity.this, menuId, "subject", text, path);

### startWrite()

#### 说明

打开留言板写文章画面。

#### 文章版面

public static void startWrite(Activity activity, int menuId, String subject, String text);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |
| menuId | int | N | 留言板ID（基本值：-1）。留言板ID是NAVER Cafe留言板UR中menuid参数的值。 |
| subject | String | N | 文章标题 |
| text | String | N | 文章正文 |

返回值

无

代码例子

// 填入基本标题, 正文开启写文章画面。

如是int menuId = 4; // 0，则不能选择菜单。

String text = " 填入基本标题, 正文开启写文章画面。";

Glink.startWrite(MainActivity.this, "subject", text);

### stop()

#### 说明

立即关闭NAVER Cafe SDK画面，终止NAVER Cafe SDK。

#### 文章版面

public static void stop(final Activity activity);

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| activity | Activity | Y | 实行方法的活动的Context客体 |

返回值

无

代码例子

// 用stop()方法终止。

Glink.stop(activity);

# 用于 iOS应用程序的说明

## 开发环境

### 软件要求事项

#### 开发工具

* IDE: Xcode 6.0以上

注意

ARC（automatic reference counting）适用于资源库。

#### 资源库

如想使用NAVER Cafe SDK，需将下面的资源库添加到项目中一起建立。

表 2 使用iOS用NAVER Cafe SDK必需的资源库

|  |  |
| --- | --- |
| 资源库 | 下载URL |
| NAVER Login | * 包含在NAVER Cafe SDK资源库文件中（4.0.6版本）。 * 下载 URL: <https://static.nid.naver.com/images/web/devcenter/3rdparty_login_library_ios_4.0.6.zip> |
| AFNetworking 1.0 以上 | * 包含在NAVER Cafe SDK资源库文件中（2.6.1版本）。 * 下载 URL: <https://github.com/AFNetworking/AFNetworking> |

### 资源库构成

iOS用NAVER Cafe SDK资源库如下构成。

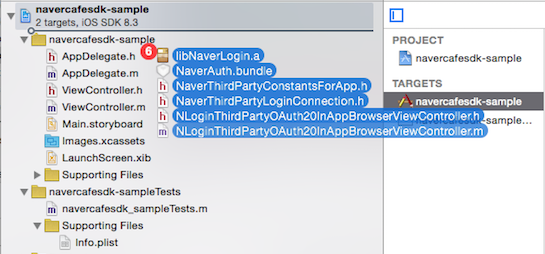
* lib: NAVER Cafe SDK资源库文件夹
  + NaverCafeSDK.bundle
  + NaverCafeSDK.framework
* sample: 使用NAVER Cafe SDK资源库的示例项目文件夹和必需的资源库文件夹
  + external-lib: NAVER Login资源库和AFNetworking资源库文件夹
  + navercafesdk-sample-ios: NAVER Cafe SDK示例项目文件夹

### 设置开发环境

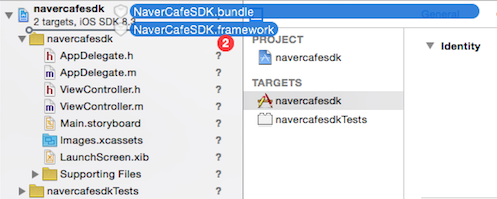
#### 设置Xcode

如想使用NAVER Cafe SDK，就在Xcode中如下设置开发环境。

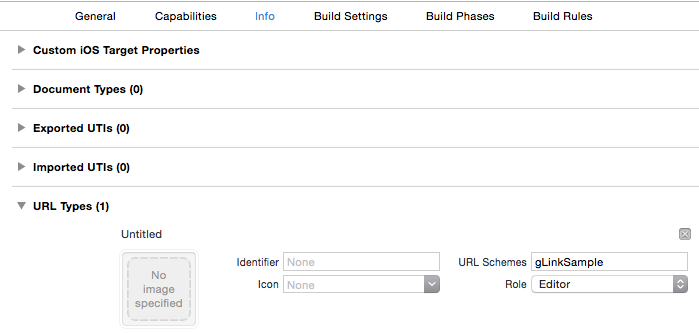
* + - * 1. 解除对NAVER Login资源库的压缩。
        2. 将NAVER ID登录资源库从Xcode添加到项目中。



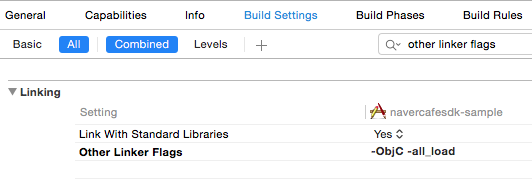
* + - * 1. 将AFNetworking资源库从Xcode中添加到项目中。
        2. 解除对NAVER Cafe SDK资源库的压缩。

将NaverCafeSDK.framework和NaverCafeSDK.bundle从Xcode中添加到项目中。 

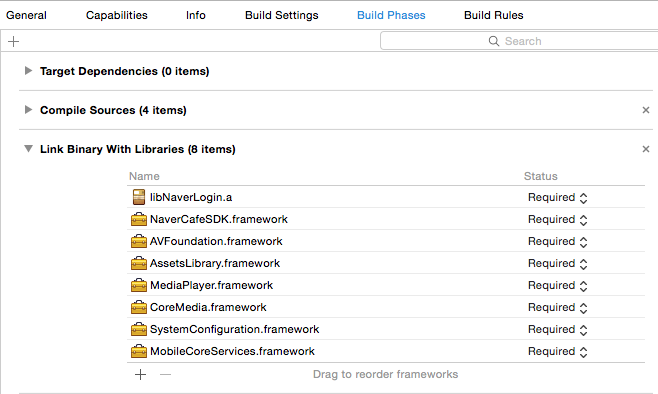
* + - * 1. 在NAVER Login 上登录应用程序时输入的URL Scheme登录到了项目中。



* + - * 1. 为了能够使用静态资源库，在Build Settings的Other Linker Flags中设置 -ObjC –all\_load选项。



* + - * 1. 在Build Phases的Link Binary With Libraries中添加如下资源库。
  + MobileCoreServices.framework
  + SystemConfiguration.framework
  + MediaPlayer.framework
  + AVFoundation.framework
  + CoreMedia.framework



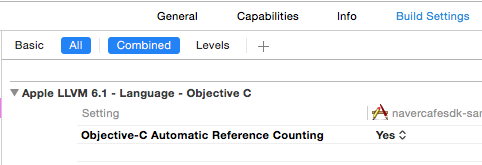
如使用Cocos2d-x引擎，要添加如下资源库。

* + MobileCoreServices.framework
  + SystemConfiguration.framework
  + MediaPlayer.framework
  + AVFoundation.framework
  + CoreMedia.framework
  + GameController.framework
  + AssetsLibrary.framework
  + Security.framework

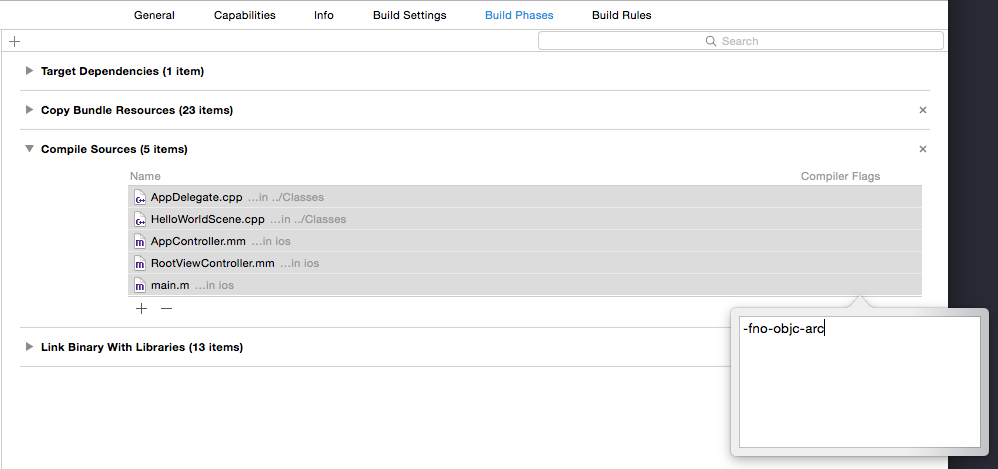
#### MRC环境转换为 ARC环境

使用Cocos2d-x引擎的项目如果是MRC（manual reference counting）环境，要用如下方法转换为ARC（automatic reference counting）环境。

* + - * 1. 将Build Settings的Objective-C Automatic Reference Counting设置为YES。



* + - * 1. 从Build Phases的Compile Sources要编译文件的Compile Flags设置为-fno-objc-arc。



## 使用NAVER Cafe SDK

### NAVER Cafe SDK初始化

用如下方式初始化NAVER Cafe SDK。客户端ID和客户端密码是 NAVER Login 应用程序得到的值。

//ViewController

#import <NaverCafeSDK/NCSDKManager.h>

- (void)viewDidLoad {

//使用 NAVER Login 的登录信息, Cafe ID初始化NAVER Cafe SDK。

[[NCSDKManager getSharedInstance] setNaverLoginConsumerKey:@"客户端ID"

naverLoginConsumerSecret:@"客户端密码"

cafeId:00000000];

}

### 设置 NAVER Login

如下设置 AppDelegate，如用NAVER ID完成登录就在NAVER Cafe SDK设置登录信息。

//AppDelegate

#import <NaverCafeSDK/NCSDKLoginManager.h>

- (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation {

//在 NAVER Login 的客体上设置登录信息。

return [[NCSDKLoginManager getSharedInstance] finishNaverLoginWithURL:url];

}

### 开启NAVER Cafe SDK 画面

如想开启NAVER Cafe SDK画面，就要设置NAVER Cafe SDK View Controller实行presentMainViewController方法。

#import <NaverCafeSDK/NCSDKManager.h>

//运行Cafe主页。

[[NCSDKManager getSharedInstance] setParentViewController:self];

[[NCSDKManager getSharedInstance] presentMainViewController];

如替代presentMainViewController方法使用presentMainViewControllerWithTabIndex方法，就可以以 NAVER Cafe SDK 画面的特定标签页状态开启NAVER Cafe SDK画面。

如使用presentMainViewControllerWithArticleId方法，可以以选定的留言板文章打开的状态开启NAVER Cafe SDK画面。

### 写文章

如打开在Cafe留言板写文章的画面，就要如下使用presentArticlePostViewControllerWithMenuId方法。

//运行写文章画面。

[[NCSDKManager getSharedInstance] setParentViewController:self];

[[NCSDKManager getSharedInstance] presentArticlePostViewControllerWithMenuId:0 subject:@"文章标题" content:@"文章内容" filePath:@"document"];

## API 参照

### NCSDKManager

调控NAVER Cafe SDK的Class。使用NCSDKManager Class的方法实现初始化或是开启、终止NAVER Cafe SDK的功能。运行NAVER Cafe SDK的最高级ViewController parentViewController具有Class。

NCSDKManager Class的方法如下。

* dismissViewController
* dismissTopViewController
* getSharedInstance
* navercafeRootViewController
* resetSharedInstance
* presentArticlePostViewControllerWithMenuId
* presentArticlePostViewControllerWithType
* presentMainViewController
* presentMainViewControllerWithArticleId
* presentMainViewControllerWithTabIndex
* presentViewController
* setGameUserId
* setNaverLoginClientId

### @property (nonatomic, weak) id parentViewController

#### 说明

运行NAVER Cafe SDK的 ViewController。运行NAVER Cafe SDK的父类。

#### 文章版面

@property (nonatomic, weak) id parentViewController;

参数

无

返回值

UIViewController客体

代码例子

[[NCSDKManager getSharedInstance] setParentViewController:self]

### (void)dismissViewController

#### 说明

删除在运行NAVER Cafe SDK的ViewController上面被运行的其他ViewController。

#### 文章版面

- (void)dismissViewController:(id)viewController;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| viewController | id | Y | 在NAVER Cafe SDK ViewController上面运行的ViewController |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] dismissViewController:self];

### (void)dismissTopViewController

#### 说明

删除在运行NAVER Cafe SDK的ViewController上面运行的其他ViewController中位于最上面的　ViewController。

#### 文章版面

- (void)dismissTopViewController;

参数

无

返回值

无

代码例子

[NCSDKManager getSharedInstance] dismissTopViewController];

### (NCSDKManager \*)getSharedInstance

#### 说明

获取NAVER Cafe SDK Instance（单例Instance）。

#### 文章版面

+ (NCSDKManager \*)getSharedInstance;

参数

无

返回值

NCSDKManager客体

代码例子

[NCSDKManager getSharedInstance]

### (id)navercafeRootViewController

#### 说明

获取NAVER Cafe SDK最上端的ViewController客体。

#### 文章版面

- (id)navercafeRootViewController;

参数

无

返回值

UIViewController客体

代码例子

[[NCSDKManager getSharedInstance] navercafeRootViewController]

### (void)resetSharedInstance

#### 说明

删除NCSDKManager客体。

#### 文章版面

+ (void)resetSharedInstance;

参数

无

返回值

无

代码例子

[NCSDKManager resetSharedInstance]

### (void)presentArticlePostViewControllerWithMenuId

#### 说明

打开留言板写文章画面。

#### 文章版面

- (void)presentArticlePostViewControllerWithMenuId:(NSInteger)menuId

subject:(NSString \*)subject

content:(NSString \*)content;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| menuId | NSInteger | Y | 留言板ID（基本值：0）。留言板ID是 NAVER Cafe 留言板URL中menuid参数的值。 |
| subject | NSString | N | 文章标题 |
| content | NSString | N | 文章正文 |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] presentArticlePostViewControllerWithMenuId:1

subject:@"标题"

content:@"内容"];

### (void)presentArticlePostViewControllerWithType

#### 说明

附加文件的状态下打开留言板写文章画面。

#### 文章版面

- (void)presentArticlePostViewControllerWithType:(GLArticlePostType)type

menuId:(NSInteger)menuId

subject:(NSString \*)subject

content:(NSString \*)content

filePath:(NSString \*)filePath;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| type | GLArticlePostType | Y | 附加文件类型   * 1: 图片文件 * 2: 视频文件 |
| menuId | NSInteger | Y | 留言板ID（基本值：0）。留言板ID是 NAVER Cafe 留言板URL中menuid参数的值。 |
| subject | NSString | N | 文章标题 |
| content | NSString | N | 文章正文 |
| filePath | NSString | Y | 附加文件径路 |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] presentArticlePostViewControllerWithType:kGLArticlePostTypeVideo

menuId:1

subject:@"标题"

content:@"内容"

filePath:@"private/var/mobile/Applications/0D1657F9-EACF-4D64-BC8A-4E01EB4FF247/tmp/trim.2CC623C7-78C3-4597-BA75-9BA12BFEF333.MOV"];

### (void)presentMainViewController

#### 说明

打开NAVER Cafe SDK画面。

#### 文章版面

- (void)presentMainViewController;

参数

无

返回值

无

代码例子

[[NCSDKManager getSharedInstance] presentMainViewController];

### (void)presentMainViewControllerWithArticleId

#### 说明

以打开留言板文章的状态打开NAVER Cafe SDK画面。

#### 文章版面

- (void)presentMainViewControllerWithArticleId:(NSUInteger)articleId;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| articleId | NSUInteger | Y | 留言ID（基本值：0） |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] presentMainViewControllerWithArticleId:10];

### (void)presentMainViewControllerWithTabIndex

#### 说明

以选定制定标签页的状态打开NAVER Cafe SDK画面。

#### 文章版面

- (void)presentMainViewControllerWithTabIndex:(NSUInteger)tabIndex;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| tabIndex | NSUInteger | Y | 要选的标签页索引值（基本值：0）   * 0: 主标签页 * 1: 公告标签页 * 2: 活动标签页 * 3: 统合留言板标签页 * 4: 个人资料标签页 |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] presentMainViewControllerWithTabIndex:1];

### (void)presentViewController

#### 说明

在NAVER Cafe SDK最上端的ViewController中运行其他ViewController。

#### 文章版面

- (void)presentViewController:(id)viewController;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| viewController | id | Y | ViewController客体的ID |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] presentViewController:self];

### (void)setGameUserId

#### 说明

关联用户的游戏ID和NAVER ID，在个人资料画面标识出ID。

#### 文章版面

- (void)setGameUserId:(NSString \*)gameUserId fieldName:(NSString \*)fieldName;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| gameUserId | NSString | Y | 用户的游戏ID |
| fieldName | NSString | N | 要在个人资料画面标识的ID（基本值: 游戏ID） |

返回值

无

代码例子

[[NCSDKManager getSharedInstance] setGameUserId:@"abc3251235" fieldName:@"游戏ID"];

### (void)setNaverLoginClientId

#### 说明

设置 NAVER Login 的客体。

#### 文章版面

- (void)setNaverLoginClientId:(NSString \*)naverLoginClientId

naverLoginClientSecret:(NSString \*)naverLoginClientSecret

cafeId:(NSInteger)cafeId;

参数

无

返回值

无

代码例子

[[NCSDKManager getSharedInstance] setNaverLoginConsumerKey:@"Consumer ID"

naverLoginConsumerSecret:@"Secret ID"

cafeId:00000000];

### NCSDKLoginManager

在NAVER Cafe SDK中调控 NAVER Login 功能的Class。

NCSDKLoginManager Class的方法如下。

* accessToken
* accessTokenExpireTime
* finishNaverLoginWithURL
* getSharedInstance
* isLoginWithFinish
* isValidAccessTokenExpireTimeNow
* loginWithFinish
* logout
* refreshAccessToken
* refreshAccessTokenWithFinish
* requestDeleteToken
* setIsInAppOauthEnable
* setIsNaverAppOauthEnable

### @property (nonatomic, weak) UIViewController \*rootViewController

#### 说明

运行登录UIView的ViewController

#### 文章版面

@property (nonatomic, weak) UIViewController \*rootViewController;

参数

无

返回值

UIViewController客体

代码例子

[[NCSDKLoginManager getSharedInstance] setRootViewController:self];

### (NSString \*)accessToken

#### 说明

NAVER登录认证后回收在NAVER SERVER中获取的访问令牌（access token）。

#### 文章版面

- (NSString \*)accessToken;

参数

无

返回值

访问令牌

代码例子

[[NCSDKLoginManager getSharedInstance] accessToken];

### (NSString \*)accessTokenExpireTime

#### 说明

确认访问令牌（access token）的终结时间。

#### 文章版面

- (NSString \*)accessTokenExpireTime;

参数

无

返回值

访问令牌的终结时间

代码例子

[[NCSDKLoginManager getSharedInstance] accessTokenExpireTime];

### (BOOL)finishNaverLoginWithURL

#### 说明

NAVER Login 完成后运行AppDelegate。

#### 文章版面

- (BOOL)finishNaverLoginWithURL:(NSURL \*)url;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| url | NSURL | Y | 完成 NAVER Login 后用AppDelegate可回拨的URL Scheme |

返回值

* true: 登录成功
* false: 登录失败

代码例子

- (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation {

return [[NCSDKLoginManager getSharedInstance] finishNaverLoginWithURL:url];

}

### (NCSDKLoginManager \*)getSharedInstance

#### 说明

NAVER Login 获取Instance（单例Instance）。

#### 文章版面

+ (NCSDKLoginManager \*)getSharedInstance;

参数

无

返回值

NCSDKLoginManager 客体

代码例子

[NCSDKLoginManager getSharedInstance]

### (void)isLoginWithFinish

#### 说明

确认NAVER登录认证后在NAVER SERVER中获取的访问令牌（access token）是否有效。

#### 文章版面

- (void)isLoginWithFinish:(void (^)(BOOL successACToken))finish;

参数

无

返回值

* true: 有效的访问令牌。
* false: 访问令牌过期，是无效的访问令牌。

代码例子

[[NCSDKLoginManager getSharedInstance] isLoginWithFinish:^(BOOL successACToken) {

if (successACToken) {

}

}];

### (BOOL)isValidAccessTokenExpireTimeNow

#### 说明

确认是否有访问令牌（access token）以及有效时间剩余多少。但是，在SERVER中有效期已过的情况下不得确认。

#### 文章版面

- (BOOL)isValidAccessTokenExpireTimeNow;

参数

无

返回值

* true: 有效的访问令牌。
* false: 访问令牌过期，是无效的访问令牌。

代码例子

[[NCSDKLoginManager getSharedInstance] isValidAccessTokenExpireTimeNow];

### (void)loginWithFinish

#### 说明

开始 NAVER Login 进行认证过程，并获取退回的访问令牌（access token）。

#### 文章版面

- (void)loginWithFinish:(void (^)(BOOL successACToken))finish;

参数

无

返回值

* true: 有效的访问令牌。
* false: 访问令牌过期，是无效的访问令牌。

代码例子

[[NCSDKLoginManager getSharedInstance] isLoginWithFinish:^(BOOL successACToken) {

if (successACToken) {

}

}];

### (void)logout

#### 说明

删除保存在客户端的访问令牌（access token）和刷新令牌（refresh token）退出。

#### 文章版面

- (void)logout;//delete local accesstoken

参数

无

返回值

无

代码例子

[[NCSDKLoginManager getSharedInstance] logout];

### (void)refreshAccessToken

#### 说明

在NAVER登录认证后，用在NAVER SERVER中获取的刷新令牌（refresh token）更新访问令牌（access token）。

#### 文章版面

- (void)refreshAccessToken;

参数

无

返回值

无

代码例子

[[NCSDKLoginManager getSharedInstance]refreshAccessToken];

### (void)refreshAccessTokenWithFinish

#### 说明

NAVER登录认证后在NAVER SERVER获取退回的刷新令牌（refresh token）。

#### 文章版面

- (void)refreshAccessTokenWithFinish:(void (^)(BOOL successACToken))finish;

参数

无

返回值

* true: 访问令牌更新成功
* false: 访问令牌更新失败

代码例子

[[NCSDKLoginManager getSharedInstance] refreshAccessTokenWithFinish:^(BOOL successACToken) {

if (successACToken) {

}

}];

### (void)requestDeleteToken

#### 说明

删除保存在客户端和SERVER中的访问令牌（access token）和刷新令牌（referesh token）。

#### 文章版面

- (void)requestDeleteToken;//delete server authorization

参数

无

返回值

无

代码例子

[[NCSDKLoginManager getSharedInstance] requestDeleteToken];

### (void)setIsInAppOauthEnable

#### 说明

运行IN APP浏览器，进行登录步骤。

#### 文章版面

- (void)setIsInAppOauthEnable:(BOOL)enable;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| enable | BOOL | Y | 是否运行IN APP浏览器 |

返回值

无

代码例子

[[NCSDKLoginManager getSharedInstance] setIsInAppOauthEnable:YES];

### (void)setIsNaverAppOauthEnable

#### 说明

运行NAVER APP后，进行登录步骤。

#### 文章版面

- (void)setIsNaverAppOauthEnable:(BOOL)enable;

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| enable | BOOL | Y | 是否运行NAVER APP |

返回值

无

代码例子

[[NCSDKLoginManager getSharedInstance] setIsNaverAppOauthEnable:YES];

# 用于Unity 应用程序的说明

## 开发环境

### 软件要求事项

#### 开发工具

* 游戏引擎：Unity 4以上

### 资源库构成

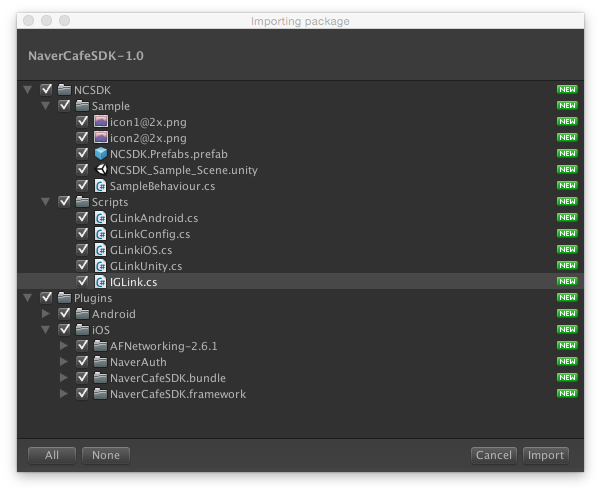
* NaverCafeSDK-1.0.unitypackage:Unity用NAVER Cafe SDK资源库

### 设置开发环境

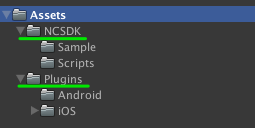
如想使用NAVER Cafe SDK，如下设置开发环境。

#### 设置Unity

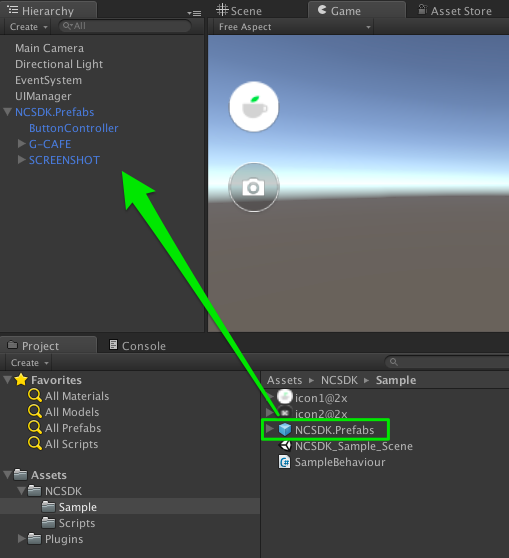
* + - * 1. 点击Unity菜单中的 Asset > Import package后，在Custom Package对话框中选择下载的 NaverCafeSDK-1.0.unityPackage文件。



* + - * 1. 在Custom Package对话框中点击Import，就会生成NCSDK文件夹和Plugins文件夹。



* + - * 1. 把在NCSDK文件夹的Sample文件夹中想要的NCSDK.Prefabs拉入其中。在NAVER Cafe SDK中生成默认提供的按钮。



* + - * 1. 在NCSDK文件夹里Scripts文件夹的GLinkConfig文件中输入Cafe ID、客户端ID和客户端密码。

//NCSDK/Scripts/GLinkConfig

static class GLinkConfig

{

public const string NaverLoginConsumerKey =

"Consumer ID";

public const string NaverLoginConsumerSecret =

"Secret ID";

public const int CafeId =

00000000;

}

#### 设置Android

使用Unity 5引擎时，如下添加NAVER Cafe SDK资源库。

* 将NAVER Cafe SDK资源库文件（cafeSdk-x.x.x.aar文件）和必需资源库文件添加到Assets/Plugins/Android文件夹。

使用Unity 4引擎时，在Eclipse中需要如下的添加设置。

* + - * 1. 解除获取的NAVER Cafe SDK资源库文件夹中cafeSdk-x.x.x.zip文件的压缩。
        2. 在解除了压缩的NAVER Cafe SDK资源库文件夹下的libs文件夹中添加构建时必需的资源库文件。
        3. 在Eclipse中解除压缩的文件夹导入到项目。
        4. 在项目设置对话框中点击Android，选择Is Library。
        5. 在AndroidManifest.xml文件中如下添加在NAVER Cafe SDK中使用的活动。

<activity

android:name="com.nhn.android.naverlogin.ui.OAuthLoginActivity"

android:screenOrientation="sensorLandscape"

android:theme="@android:style/Theme.Translucent.NoTitleBar"/>

<activity

android:name="com.nhn.android.naverlogin.ui.OAuthLoginInAppBrowserActivity"

android:screenOrientation="sensorLandscape"

android:label="OAuth2.0 In-app"/>

<activity

android:name=".ui.VideoPlayActivity"

android:screenOrientation="sensorLandscape"

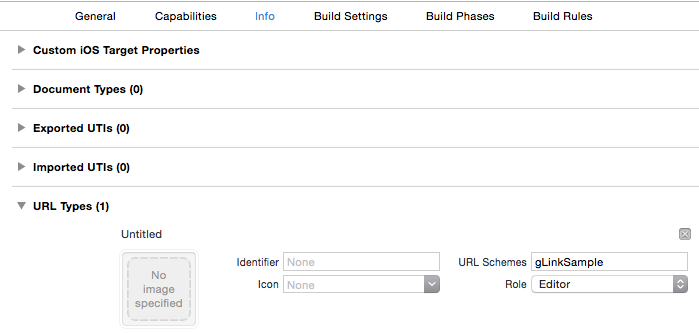
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>

* + - * 1. 构建相应的项目。
        2. 将用Eclipse构建的文件添加到Unity项目的Assets/Plugins/Android文件夹中。
        3. 构建Unity项目。

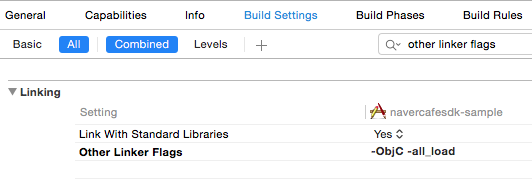
#### 设置iOS

开发iOS应用程序时如下设置Xcode。

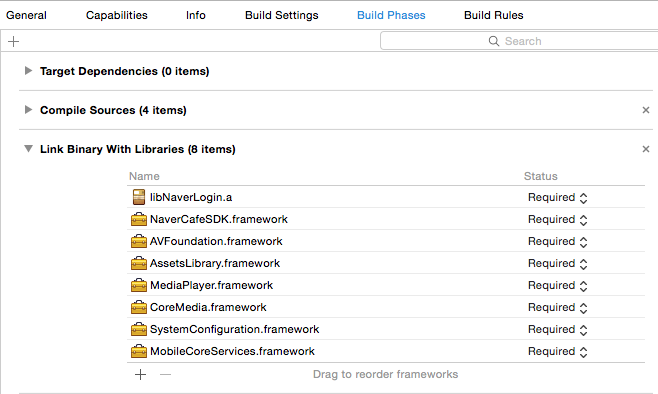
* + - * 1. 在NAVER Login上登录应用程序时，将输入的URL Scheme登录到Xcode项目中。

1

* + - * 1. 为了能够使用静态资源库，在Build Settings的Other Linker Flags中设置-ObjC –all\_load选项。



* + - * 1. 在Build Phases的Link Binary With Libraries中添加下面的资源库。
  + MobileCoreServices.framework
  + SystemConfiguration.framework
  + MediaPlayer.framework
  + AVFoundation.framework
  + CoreMedia.framework
  + Security.framework
  + AssetsLibrary.framework



* + - * 1. 如下设置App Delegate，使 NAVER Login 完成后App Delegate被呼出。

//AppDelegate

# import <NaverCafeSDK/NCSDKLoginManager.h>

- (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation {

return [[NCSDKLoginManager getSharedInstance] finishNaverLoginWithURL:url];

}

## 使用NAVER Cafe SDK

### 运行

如想开启NAVER Cafe SDK画面，要如下呼出GlinkUnity.executeMain()方法。

// Cafe 主页

GlinkUnity.executeMain ();

游戏玩家在Cafe写文章的画面如下用 GlinkUnity.executeArticlePostWithImgae()方法实现。

//截屏

GlinkUnity.executeArticlePostWithImgae(menuId, "留言标题", "留言内容", image path);

如使用GlinkUnity .executeArticlePostWithImage()方法或是GlinkUnity .executeArticlePostWithVideo()方法，可以在附加图片或是视频的状态下打开留言板写文章画面。

### 映射用户的游戏ID和 NAVER ID

NAVER Cafe SDK提供映射用户的游戏ID和NAVER ID的功能。如在PC上接触Cafe管理者页面，可以确认以影射的游戏ID和NAVER ID的目录。

下面是在iOS应用程序中实现游戏ID和NAVER ID映射功能的例子。

/\*

为了联动NAVER ID和游戏ID，而联动用户的游戏ID。

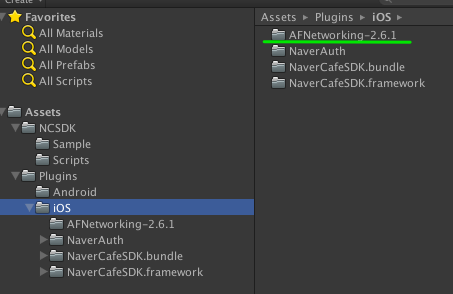
fieldName只是要在简历方面标识的字符串(基本值: 游戏ID)

\*/

- (void)setGameUserId:(NSString \*)gameUserId fieldName:(NSString \*)fieldName;

### 替换AFNetworking

在NaverCafeSDK.unitypackage中默认包含用于互联网通信的资源库AFNetworking 2.6.1。根据项目的状况也可删除AFNetworking 资源库或是更改为其他版本进行构建。但是，NAVER Cafe SDK只支持AFNetworking 1.x 以上版本。



## API 参照

### GLinkUnity

调控NAVER Cafe SDK的Class。使用GLinkUnity Class的方法实现开启或打开写文章画面的功能。

GLinkUnity Class的方法如下。

* executeArticlePost()
* executeArticlePostWithImage()
* executeArticlePostWithVideo()
* executeMain()

### executeArticlePost()

#### 说明

打开留言板写文章画面。

#### 文章版面

public static void executeArticlePost(int menuId, string subject, string content) {

sharedInstance().executeArticlePost (menuId, subject, content);

}

#### 参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| menuId | int | Y | 留言板ID（基本值: -1）。留言板ID是NAVER Cafe留言板URL中menuid参数的值。 |
| subject | string | Y | 文章标题 |
| content | string | Y | 文章正文 |

#### 返回值

无

#### 代码例子

GlinkUnity.executeArticlePost(28290504, "文章标题","文章正文" );

### executeArticlePostWithImage()

#### 说明

附加图片的状态下打开留言板写文章画面。

#### 文章版面

public static void executeArticlePostWithImage(int menuId, string subject, string content, string filePath) {

sharedInstance().executeArticlePostWithImage (menuId, subject, content, filePath);

}

#### 参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| menuId | int | Y | 留言板ID（基本值: -1）。留言板ID是NAVER Cafe留言板URL中 menuid 参数的值。 |
| subject | string | Y | 文章标题 |
| content | string | Y | 文章正文 |
| filePath | string | Y | 图片文件的径路（URI形式） |

#### 返回值

无

#### 代码例子

{

GlinkUnity.executeArticlePostWithImgae(28290504, "　文章标题", "文章内容","/navercafesdk/glink.png" );

}

### executeArticlePostWithVideo()

#### 说明

附加视频的状态下打开留言板写文章画面。

#### 文章版面

public static void executeArticlePostWithVideo(int menuId, string subject, string content, string filePath) {

sharedInstance().executeArticlePostWithVideo (menuId, subject, content, filePath);

}

#### 参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必需 | 说明 |
| menuId | int | Y | 留言板ID（基本值: -1）。留言板ID是NAVER Cafe 留言板URL中menuid参数的值。 |
| subject | string | Y | 文章标题 |
| content | string | Y | 文章正文 |
| filePath | string | Y | 视频文件的径路（URI形式） |

#### 返回值

无

#### 代码例子

{

GlinkUnity.executeArticlePostWithImgae(28290504, "文章标题", "文章内容","/navercafesdk/glink.avi" );

}

### executeMain()

#### 说明

打开NAVER Cafe SDK画面。

#### 文章版面

public static void executeMain() {

sharedInstance().executeMain ();

}

#### 参数

无

#### 返回值

无

#### 代码例子

{

GlinkUnity.executeMain ();

}