|  |  |
| --- | --- |
| NAVER Cafe SDKインテグレーションガイド | |
|  |  |

|  |
| --- |
| 著作権 |

Copyright © NAVER Corp. All Rights Reserved.

本書は、NAVER(株)の知的財産であるためいかなる場合にもNAVER(株)の許可なく本書の一部または全部をコピー・転送・配布・変更して使用することはできません。

本書は、情報提供の目的に限って提供されます。NAVER(株)は、本書に収められた情報については完全性と正確性を期しておりますが、記載内容の誤りや漏れによって発生する問題については責任を負いません。

したがって、本書の使用や使用した結果に伴う責任は全面的に利用者にあり、NAVER(株)はこれについて明示的または黙示的にいかなる保証もいたしません。

関連URL情報を含め、本書で言及した特定のソフトウェア商品および製品は、当該所有者の属した現地および国内外の関連法に従い、当該法律を遵守しないことにより発生するあらゆる結果に対する責任は全面的に利用者自身にあります。

NAVER(株)は、本書の内容を予告なく変更する場合があります。

|  |
| --- |
| はじめに |

本書について

本書では、NAVER Cafe SDKをアプリケーションに組み込む方法について説明します。

読者

本書の読者は、アプリケーションへのNAVER Cafe SDK組み込み作業を行う開発者です。

お問い合わせ

本書の内容に関する問い合わせは下記までご連絡ください。

連絡先: [dl\_gamesdkpartner@navercorp.com](mailto:dl_gamesdkpartner@navercorp.com)

改版履歴

|  |  |
| --- | --- |
| 日付 | 履歴 |
| 2015年12月23日 | 初版発行 |

|  |
| --- |
| 表記規則 |

参考表記

参考

読者が参考にすべき内容を記述します。

ユーザー入力値

本書においてUI文言、ユーザーの固定・可変入力値は下記のように表記します。

* UI文言: メニュー > サブメニュー
* ユーザー固定入力値: localhostを入力します。
* ユーザー可変入力値: <http://www.naver.com/>{企業名}

ソースコードの表記

本書においてソースコードは、灰地に黒字で表記します。

COPYDATASTRUCT st;

st.dwData = PURPLE\_OUTBOUND\_ENDING;

st.cbData = sizeof(pp);

st.lpData = &pp;

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

|  |
| --- |
| 目次 |

[SDK概要 9](#_Toc438559620)

[NAVER Cafe SDKとは？ 9](#_Toc438559621)

[NAVER Cafe SDKの機能と特徴 9](#_Toc438559622)

[主要機能 9](#_Toc438559623)

[使用環境 9](#_Toc438559624)

[Android向け 11](#_Toc438559625)

[開発環境 11](#_Toc438559626)

[ソフトウェア要件 11](#_Toc438559627)

[ライブラリの構成 12](#_Toc438559628)

[開発環境セットアップ 12](#_Toc438559629)

[NAVER Cafe SDKの組み込み手順 13](#_Toc438559630)

[AndroidManifest.xmlの設定 13](#_Toc438559631)

[NAVER Cafe SDKの初期化 13](#_Toc438559632)

[NAVER Cafe SDK画面を開く 13](#_Toc438559633)

[NAVER Cafe SDK画面を閉じる 14](#_Toc438559634)

[投稿作成 14](#_Toc438559635)

[App Schemeの処理 14](#_Toc438559636)

[ユーザーのゲームIDとのマッピング 14](#_Toc438559637)

[リソース画像の変更 15](#_Toc438559638)

[APIリファレンス 15](#_Toc438559639)

[Glink 15](#_Toc438559640)

[init() 15](#_Toc438559641)

[isShowGlink() 16](#_Toc438559642)

[popBackStack() 17](#_Toc438559643)

[setGameUserId() 17](#_Toc438559644)

[startEvent() 18](#_Toc438559645)

[startImageWrite() 18](#_Toc438559646)

[startHome() 19](#_Toc438559647)

[startMenu() 19](#_Toc438559648)

[startNotice() 20](#_Toc438559649)

[startProfile() 20](#_Toc438559650)

[startVideoWrite() 21](#_Toc438559651)

[startWrite() 21](#_Toc438559652)

[stop() 22](#_Toc438559653)

[iOS向け 23](#_Toc438559654)

[開発環境 23](#_Toc438559655)

[ソフトウェア要件 23](#_Toc438559656)

[ライブラリの構成 23](#_Toc438559657)

[開発環境セットアップ 24](#_Toc438559658)

[NAVER Cafe SDKの組み込み手順 27](#_Toc438559659)

[NAVER Cafe SDKの初期化 27](#_Toc438559660)

[NAVER Loginの設定 27](#_Toc438559661)

[NAVER Cafe SDK画面を開く 27](#_Toc438559662)

[投稿作成 28](#_Toc438559663)

[APIリファレンス 28](#_Toc438559664)

[NCSDKManager 28](#_Toc438559665)

[@property(nonatomic, weak) id parentViewController 29](#_Toc438559666)

[(void)dismissViewController 29](#_Toc438559667)

[(void)dismissTopViewController 29](#_Toc438559668)

[(NCSDKManager \*)getSharedInstance 30](#_Toc438559669)

[(id)NAVER CafeRootViewController 30](#_Toc438559670)

[(void)resetSharedInstance 31](#_Toc438559671)

[(void)presentArticlePostViewControllerWithMenuId 31](#_Toc438559672)

[(void)presentArticlePostViewControllerWithType 32](#_Toc438559673)

[(void)presentMainViewController 33](#_Toc438559674)

[(void)presentMainViewControllerWithArticleId 33](#_Toc438559675)

[(void)presentMainViewControllerWithTabIndex 33](#_Toc438559676)

[(void)presentViewController 34](#_Toc438559677)

[(void)setGameUserId 34](#_Toc438559678)

[(void)setNaverLoginClientId 35](#_Toc438559679)

[NCSDKLoginManager 35](#_Toc438559680)

[@property (nonatomic, weak) UIViewController \*rootViewController 36](#_Toc438559681)

[(NSString \*)accessToken 36](#_Toc438559682)

[(NSString \*)accessTokenExpireTime 37](#_Toc438559683)

[(BOOL)finishNaverLoginWithURL 37](#_Toc438559684)

[(NCSDKLoginManager \*)getSharedInstance 38](#_Toc438559685)

[(void)isLoginWithFinish 38](#_Toc438559686)

[(BOOL)isValidAccessTokenExpireTimeNow 38](#_Toc438559687)

[(void)loginWithFinish 39](#_Toc438559688)

[(void)logout 39](#_Toc438559689)

[(void)refreshAccessToken 40](#_Toc438559690)

[(void)refreshAccessTokenWithFinish 40](#_Toc438559691)

[(void)requestDeleteToken 41](#_Toc438559692)

[(void)setIsInAppOauthEnable 41](#_Toc438559693)

[(void)setIsNaverAppOauthEnable 42](#_Toc438559694)

[Unity向け 43](#_Toc438559695)

[開発環境 43](#_Toc438559696)

[ソフトウェア要件 43](#_Toc438559697)

[ライブラリの構成 43](#_Toc438559698)

[開発環境セットアップ 43](#_Toc438559699)

[NAVER Cafe SDKの組み込み手順 47](#_Toc438559700)

[実行 47](#_Toc438559701)

[ユーザーのゲームIDとNAVER IDとのマッピング 47](#_Toc438559702)

[AFNetworkingの交替 48](#_Toc438559703)

[APIリファレンス 48](#_Toc438559704)

[GLinkUnity 48](#_Toc438559705)

[executeArticlePost() 48](#_Toc438559706)

[executeArticlePostWithImage() 49](#_Toc438559707)

[executeArticlePostWithVideo() 50](#_Toc438559708)

[executeMain() 50](#_Toc438559709)

|  |
| --- |
| 図表一覧 |

表一覧

[表 1 Android向けNAVER Cafe SDKライブラリ 11](#_Toc438559710)

[表 2 iOS向けNAVER Cafe SDKライブラリ 23](#_Toc438559711)

図一覧

[図 1 NAVER Cafe SDK 9](#_Toc438559712)

# SDK概要

## NAVER Cafe SDKとは？

NAVER Cafe SDKはゲームエンジンに関係なくモバイルゲームにNAVER Cafeを簡単に組み込めるライブラリです。NAVER Cafe SDKを組み込むことでゲームユーザーはゲームを続けたままゲームコミュニティ(NAVER Cafe)との交流が可能になります。

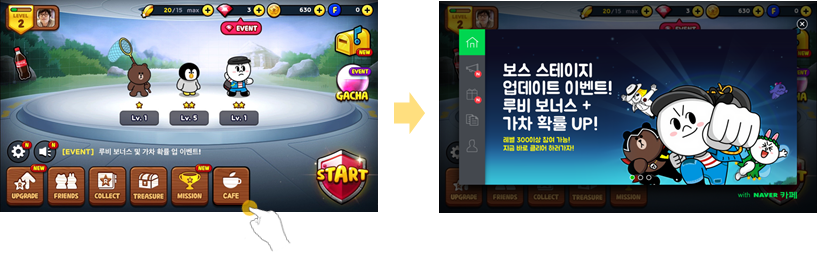


図 1 NAVER Cafe SDK

## NAVER Cafe SDKの機能と特徴

### 主要機能

NAVER Cafe SDKの主な機能は下記のとおりです。

* iOS、Androidに対応
* Unity 4以上とCocos2d-x 2.1以上のゲームエンジンに対応
* Cafe機能を利用できる画面を提供
* OAuth 2.0ベースのNAVER　Loginを用いたユーザー認証

### 使用環境

#### オペレーティングシステム

NAVER Cafe SDKは以下の環境で動作します。

* Android: Android 4.2(Jelly Bean) API Level 17以上
* iOS: iOS 7.0以上(ARMv7、ARMv7s、ARM64に対応)

#### NAVER Loginの登録

NAVER Cafe SDKにおいてNAVER Login機能を用いてユーザー認証を行うにはクライアントIDとクライアントシークレットが必要です。

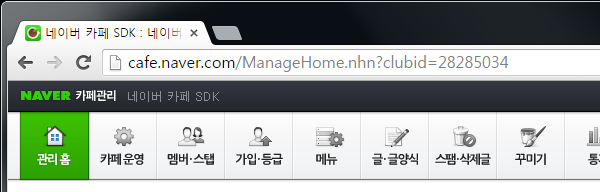
* クライアントID: NAVER Loginにゲームを登録すると発行されるクライアントID。
* クライアントシークレット: NAVER Loginにゲームを登録すると発行されるクライアントシークレット。

クライアントIDとクライアントシークレットは、NAVER Loginにアプリケーションを登録すると発行されます。NAVER Loginにアプリケーションを登録してクライアントIDとクライアントシークレットを確認する方法については、NAVER Login Developersのドキュメントを参考にするか、または担当([dl\_gamesdkpartner@navercorp.com](mailto:dl_gamesdkpartner@navercorp.com))までお問い合わせください。

* NAVER Login Developers(韓国語): https://nid.naver.com/devcenter/main.nhn

#### NAVER Cafe ID

NAVER Cafe SDKを組み込むにはNAVER CafeのCafe IDが必要です。Cafe IDは、NAVER Cafe管理ページURLのclubidパラメータ値にあたります。



# Android向け

## 開発環境

### ソフトウェア要件

#### 開発ツール

* IDE: Android Studio、またはEclipse
* Android Support Library v7

#### ライブラリ

NAVER Cafe SDKを組み込むには、下記のライブラリをプロジェクトに追加してビルドを行います。

表 1 Android向けNAVER Cafe SDKライブラリ

|  |  |
| --- | --- |
| ライブラリ | ダウンロードURL |
| NAVER Login | * NAVER Cafe SDKライブラリファイルに含まれている(4.1.4バージョン)。 * ダウンロードURL: <https://static.nid.naver.com/images/web/devcenter/3rdparty_login_library_android_4.1.4.zip> |
| NAVER Volleyer | * NAVER Cafe SDKライブラリファイルに含まれている(2.0.1バージョン)。 * ダウンロードURL: <http://mvnrepository.com/artifact/com.navercorp.volleyextensions/volleyer> |
| Volley | * NAVER Cafe SDKライブラリファイルに含まれている(1.0.2バージョン)。 * ダウンロードURL: [http://mvnrepository.com/artifact/com.mcxiaoke.volley/library](http://mvnrepository.com/artifact/com.mcxiaoke.volley/library/) |
| Google Gson | * NAVER Cafe SDKライブラリファイルに含まれている(2.3.1バージョン)。 * ダウンロードURL: <http://mvnrepository.com/artifact/com.google.code.gson/gson> |
| Glide | * NAVER Cafe SDKライブラリファイルに含まれている(3.6.1バージョン)。 * ダウンロードURL: <http://mvnrepository.com/artifact/com.github.bumptech.glide/glide> |
| Otto | * NAVER Cafe SDKライブラリファイルに含まれている(1.3.8バージョン)。 * ダウンロードURL: <http://mvnrepository.com/artifact/com.squareup/otto> |

### ライブラリの構成

Android向けNAVER Cafe SDKライブラリは、下記のように構成されています。

* lib: NAVER Cafe SDKライブラリと必須ライブラリ
  + cafeSdk-x.x.x.aar: Android Studioで使用するNAVER Cafe SDKライブラリ
  + cafeSdk-x.x.x.zip: Eclipseで使用するNAVER Cafe SDKライブラリ
  + library: NAVER Cafe SDKを使用する際に必要なライブラリフォルダ
* sample: NAVER Cafe SDKライブラリを使ったサンプルプロジェクトフォルダ
  + NAVER Cafesdk-sample-android-studio: Android Studioで使えるNAVER Cafe SDKのサンプルプロジェクトフォルダ
  + NAVER Cafesdk-sample-eclipse-master: Eclipseで使えるNAVER Cafe SDKのサンプルプロジェクトフォルダ

### 開発環境セットアップ

NAVER Cafe SDKを組み込むには下記のようにIDEごとに開発環境をセットアップします。

#### Android Studio

* + - * 1. AndroidプロジェクトのlibsフォルダにNAVER Cafe SDKライブラリファイル(cafeSdk-x.x.x.aar)をコピーします。
        2. AndroidプロジェクトのlibsフォルダにNAVER Loginライブラリファイル(naveroauthlogin-4.x.x.jar)をコピーします。
        3. メニューから、File > Project Structureをクリックし、Project StructureダイアログボックスからApp > Dependenciesをクリックして、ライブラリを追加します。または、次のようにAndroidプロジェクトのbuild.gladleファイルに直接ライブラリを追加します。

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. NAVER Cafe SDKライブラリファイル(cafeSdk-x.x.x.zip)を解凍します。
        2. 解凍したNAVER Cafe SDKライブラリフォルダ内のlibsフォルダにビルドに必要な必須ライブラリファイルを追加します。
        3. AndroidプロジェクトのAndroidManifest.xmlファイルに、下記のようにNAVER Loginで使うアクティビティとNAVER Cafe SDKで使うアクティビティを追加します。

<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" />

## NAVER Cafe SDKの組み込み手順

### AndroidManifest.xmlの設定

AndroidプロジェクトのAndroidManifest.xmlファイルに下記の内容を追加します。

packageプロパティには、NAVER Loginにアプリケーションを登録した際のAndroid Intentの設定内容を入力します。NAVER Cafe SDKをコントロールするオブジェクトであるGlinkクラスのアクセス権も設定します。

<!—packageは、NAVER Login Developersに登録したAndroid Intentと同じでなければなりません。 -->

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

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

<!-- Glinkに必要なアクセス権-->

<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の初期化

NAVER Cafe SDKを初期化する機能は、Glink.init()メソッドで実装します。

クライアントIDとクライアントシークレットは、NAVER Loginにアプリケーションを登録した際に発行された値です。

/\*\*

\* "NAVER Login Developers"の発行情報を利用してSDKの初期化処理を行います。

\* Glinkの他のメソッドを呼び出す前に、必ず初期化処理を行ってください。

\* NAVER Login Developers: https://nid.naver.com/devcenter/main.nhn

\*/

final int cafeId = 28266581;

final String consumerKey = "クライアントID";

final String consumerSecret = "クライアントシークレット";

Glink.init(consumerKey, consumerSecret, cafeId);

### NAVER Cafe SDK画面を開く

NAVER Cafe SDKの画面は5つのメニュー(タブ)で構成されています。したがって、NAVER Cafe SDKを立ち上げる際のスタート画面をどの画面にするかを定めておく必要があります。

例えば、トップ画面でスタートさせるには、下記のようにGlink.startHome()メソッドを使います。

// トップ画面でスタートさせます。

Glink.startHome(activity);

トップ画面以外のお知らせ、イベント、掲示板、プロフィールメニューでスタートさせるには、下記のメソッドを使います。

* Glink.startNotice()メソッド: お知らせメニューでスタート
* Glink.startEvent()メソッド: イベントメニューでスタート
* Glink.startMenu()メソッド: 掲示板メニューでスタート
* Glink.startProfile()メソッド: プロフィールメニューでスタート

### NAVER Cafe SDK画面を閉じる

NAVER Cafe SDK画面を閉じる機能は、Glink.stop()メソッド、またはGlink.popBackStack()メソッドで実装します。

* Glink.stop()メソッド: 即時に画面を閉じてNAVER Cafe SDKを終了します。
* Glink.popBackStack()メソッド: backStackに溜まっている画面をひとつずつ閉じます。すべての画面を閉じたらNAVER Cafe SDKを終了します。backStackに溜まる画面は投稿閲覧画面、投稿作成画面, 検索画面です。

下記は、Glink.stop()メソッドでNAVER Cafe SDK画面を閉じる例です。

// stop()メソッドで終了する。

Glink.stop(activity);

下記は、Glink.popBackStack()メソッドでNAVER Cafe SDK画面を閉じる例です。

// popBackStackで終了する。

Glink.popBackStack(activity);

### 投稿作成

ゲームユーザーがCafeで投稿を作成する画面は下記のようにGlink.startWrite()メソッドで実装します。

// 基本タイトルと内容を入れて投稿画面を実行します。

int menuId = 4; // 0ならメニューを選択しません。

String text = "基本タイトルと内容が入れて投稿画面を実行します。";

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

Glink.startImageWrite()メソッドやGlink.startVideoWrite()メソッドを使って画像や動画を添付したまま掲示板の投稿画面を実行することができます。

### App Schemeの処理

App Schemeで移動するバナー画像がある場合、下記のようにOnClickAppSchemeBannerListenerを設定するとタッチイベントが発生した際にApp Schemeを処理する機能を実装することができます。

// アプリスキームタッチリスナーの設定

Glink.setOnClickAppSchemeBannerListener(new Glink.OnClickAppSchemeBannerListener() {

@Override public void onClickAppSchemeBanner(String appScheme) {

// Cafe管理で設定したApp Schemeの文字列をNAVER Cafe SDKから渡します。

// App Schemeを処理するコードを実装します。

}

});

### ユーザーのゲームIDとのマッピング

ユーザーのゲームIDとNAVER　IDをマッピングする機能は、下記のようにGlink.setGameUserId()メソッドで実装します。

// ユーザのゲームIDとNAVER　IDをマッピングします。

// プロフィール画面からマッピングされたゲームIDを確認することができます。

Glink.setGameUserId(this, "gameUserId", "ゲームID");

### リソース画像の変更

NAVER Cafe SDKライブラリに含まれたリソース画像は、下記のように変更できます。

* + - * 1. NAVER Cafe SDKライブラリファイル(.aaaファイル)を解凍します。
        2. 解凍したフォルダの/res/drawable-xhdpiフォルダにある画像を希望の画像に変更します。
        3. 解凍したフォルダを再度NAVER Cafe SDKライブラリファイル(.aaaファイル)に圧縮します。
        4. 再圧縮したNAVER Cafe SDKライブラリを適用してプロジェクトをビルドします。

注意

画像を変更する際の画像サイズは、元画像のサイズと同じサイズでなければなりません。

## APIリファレンス

### Glink

NAVER Cafe SDKのコントロールクラス。Glinkクラスのメソッドを使ってNAVER Cafe SDKを初期化したり、開始、中止する機能を実装します。Glinkクラスのメソッドは、下記のとおりです。

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

### init()

#### 説明

NAVER Cafe SDKの初期化を行います。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| clientId | String | Y | クライアントID。NAVER Loginにアプリケーションを登録した際に発行される値です。 |
| clientSecret | String | Yｄ | クライアントシークレット。NAVER Loginにアプリケーションを登録した際に発行される値です。 |
| cafeId | int | Y | Cafe ID。NAVER Cafe管理ページURLのclubid パラメータ値にあたります。 |

#### 返却値

なし

#### サンプルコード

//  NAVER Login情報とCafe IDを用いてNAVER Cafe SDKの初期化処理を行います。

Glink.init(“abcd”, "aaaa", 33);

### isShowGlink()

#### 説明

NAVER Cafe SDK画面の開閉有無を確認します。

#### 構文

public static boolean isShowGlink(Activity activity);

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

* true: NAVER Cafe SDK画面が開いている。
* false: NAVER Cafe SDK画面が開いていない。

サンプルコード

// NAVER Cafe SDK画面の開閉有無を確認します。

// 返却値がtrueなら開いている状態、falseなら開いていない状態です。

Glink.isShowGlink(this);

### popBackStack()

#### 説明

backStackに溜まっている画面を1つずつ閉じます。すべての画面を閉じるとNAVER Cafe SDKを終了します。 backStackに溜まる画面は投稿閲覧画面、投稿作成画面、検索画面です。

#### 構文

public static void popBackStack(Activity activity);

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

なし

サンプルコード

// NAVER Cafe SDK画面を1つずつ閉じます。

Glink.popBackStack (activity);

### setGameUserId()

#### 説明

ユーザーのゲームIDとCafe IDをマッピングします。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |
| gameUserId | String | N | ユーザーのゲームID |
| fieldName | String | N | プロフィール画面に表示するID(基本値: ユーザーのゲームID) |

#### 返却値

なし

#### サンプルコード

// ユーザーのゲームIDとNAVER IDをマッピングします。

// プロフィール画面からマッピングされたゲームIDを確認することができます。

Glink.setGameUserId(this, "gameUserId", "ゲームID");

### startEvent()

#### 説明

イベントメニューが選択された状態でNAVER Cafe SDKを立ち上げます。

#### 構文

public static void startEvent(Activity activity);

#### パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

なし

サンプルコード

// イベントメニューが選択された状態でNAVER Cafe SDKを立ち上げます。

Glink.startEvent(activity);

### startImageWrite()

#### 説明

画像を添付して掲示板の投稿作成画面を開きます。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |
| menuId | int | N | 掲示板のID(基本値: -1)。掲示板のIDは、NAVER Cafe 掲示板URLのmenuidパラメータ値にあたります。 |
| subject | String | N | 投稿のタイトル |
| text | String | N | 投稿の内容 |
| imageUri | String | N | 画像ファイルのパス(URI形式) |

返却値

なし

サンプルコード

// 基本タイトルと内容が入力されていて、画像が添付されている状態で投稿作成画面を立ち上げます。画像パスは、URI形式で入力してください。

int menuId = 4; // 0ならメニューを選択しません。

String text = "基本タイトルと内容が入力されていて、画像が添付されている状態で投稿作成画面を立ち上げます。\n画像パスは、URI形式で入力してください。";

String path = "your image uri";

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

### startHome()

#### 説明

トップ画面でNAVER Cafe SDKを立ち上げます。

#### 構文

public static void startHome(Activity activity);

#### パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

#### 返却値

なし

#### サンプルコード

// トップ画面でNAVER Cafe SDKを立ち上げます。

Glink.startHome(activity);

### startMenu()

#### 説明

掲示板が選択された状態でNAVER Cafe SDK画面を開きます。

#### 構文

public static void startMenu(Activity activity);

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

なし

サンプルコード

// 掲示板が選択された状態でNAVER Cafe SDK画面を開きます。

Glink.startMenu(activity);

### startNotice()

#### 説明

お知らせが選択された状態でNAVER Cafe SDK画面を開きます。

#### 構文

public static void startNotice(Activity activity);

#### パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

なし

サンプルコード

// お知らせが選択された状態でNAVER Cafe SDK画面を開きます。

Glink.startNotice(activity);

### startProfile()

#### 説明

プロフィールを選択した状態でNAVER Cafe SDK画面を開きます。

#### 構文

public static void startProfile(Activity activity);

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

なし

サンプルコード

// プロフィールが選択された状態でNAVER Cafe SDK画面を開きます。

Glink.startProfile(activity);

### startVideoWrite()

#### 説明

動画を添付して掲示板の投稿作成画面を開きます。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |
| menuId | int | N | 掲示板のID(基本値: -1)。掲示板のIDは、NAVER Cafe掲示板URLのmenuidパラメータ値にあたります。 |
| subject | String | N | 投稿のタイトル |
| text | String | N | 投稿の内容 |
| videoUri | String | N | 動画ファイルのパス(URI形式) |

返却値

なし

サンプルコード

// 基本タイトルと内容が入力されていて、動画が添付されている状態で投稿画面を立ち上げます。動画ファイルのパスは、URI形式で入力してください。

int menuId = 4; // 0ならメニューを選択しません。

String text = "基本タイトルと内容が入力されていて、動画が添付されている状態で投稿作成画面を起動します。\n動画ファイルのパスは、URI形式で入力してください。";

String path = "your video uri";

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

### startWrite()

#### 説明

掲示板の投稿作成画面を開きます。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |
| menuId | int | N | 掲示板のID(基本値: -1)。掲示板のIDは、NAVER Cafe掲示板URLのmenuidパラメータ値にあたります。 |
| subject | String | N | 投稿のタイトル |
| text | String | N | 投稿の内容 |

返却値

なし

サンプルコード

// 基本タイトル、内容を入れて投稿作成画面を立ち上げます。

int menuId = 4; // 0なら、メニューを選択しません。

String text = "基本タイトル、内容を入れて投稿作成画面を立ち上げます。";

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

### stop()

#### 説明

NAVER Cafe SDK画面を即時に閉じ、NAVER Cafe SDKを終了します。

#### 構文

public static void stop(final Activity activity);

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| activity | Activity | Y | メソッドを実行するアクティビティのContextオブジェクト |

返却値

なし

サンプルコード

// stop()メソッドで終了します。

Glink.stop(activity);

# iOS向け

## 開発環境

### ソフトウェア要件

#### 開発ツール

* IDE: Xcode 6.0以上

注意

ライブラリにARC(automatic reference counting)が適用されています。

#### ライブラリ

NAVER Cafe SDKを組み込むには下記のライブラリをプロジェクトに追加してビルドします。

表 2 iOS向けNAVER Cafe SDKライブラリ

|  |  |
| --- | --- |
| ライブラリ | ダウンロードURL |
| NAVER Login | * NAVER Cafe SDKライブラリファイルに含まれています(4.0.6バージョン)。 * ダウンロードURL: <https://static.nid.naver.com/images/web/devcenter/3rdparty_login_library_ios_4.0.6.zip> |
| AFNetworking 1.0以上 | * NAVER Cafe SDKライブラリファイルに含まれています(2.6.1バージョン)。 * ダウンロードURL: <https://github.com/AFNetworking/AFNetworking> |

### ライブラリの構成

iOS向けNAVER Cafe SDKライブラリは、下記のように構成されています。

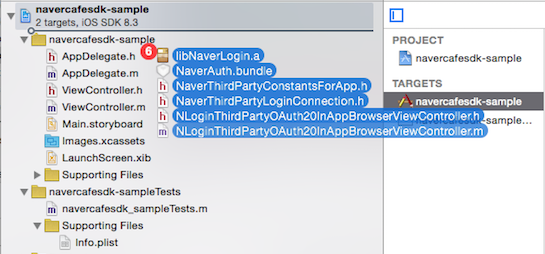
* lib: NAVER Cafe SDKライブラリフォルダ
  + NAVER CafeSDK.bundle
  + NAVER CafeSDK.framework
* sample: NAVER Cafe SDKライブラリを使用するサンプルプロジェクトと必須ライブラリフォルダ
  + external-lib: NAVER LoginライブラリとAFNetworkingライブラリフォルダ
  + NAVER Cafesdk-sample-ios: NAVER Cafe SDKサンプルプロジェクトフォルダ

### 開発環境セットアップ

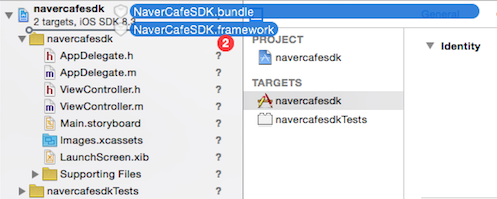
#### Xcodeの設定

NAVER Cafe SDKを組み込むにはXcodeにて下記のように開発環境をセットアップします。

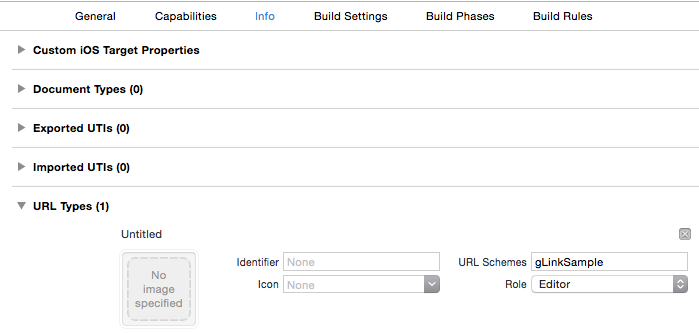
* + - * 1. NAVER Loginライブラリを解凍します。
        2. XcodeでNAVER Loginライブラリをプロジェクトに追加します。



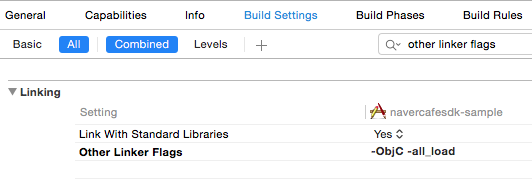
* + - * 1. XcodeでAFNetworkingライブラリをプロジェクトに追加します。
        2. NAVER Cafe SDKライブイラリを解凍します。
        3. XcodeでNAVER CafeSDK.frameworkとNAVER CafeSDK.bundleをプロジェクトに追加します。



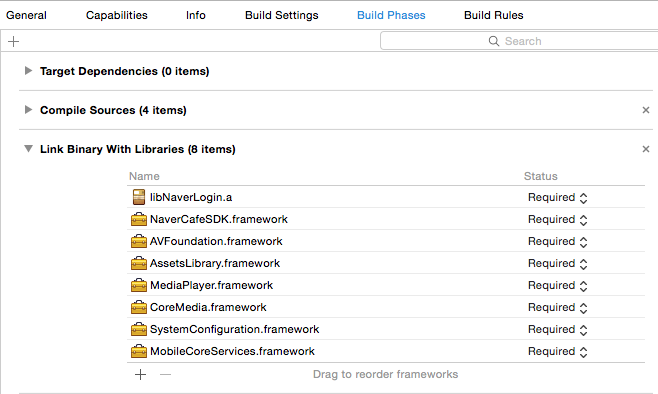
* + - * 1. NAVER Loginでアプリケーション登録時に入力したURL Schemeをプロジェクトに登録します。



* + - * 1. スタティックライブラリを使えるようにBuild SettingsのOther Linker Flagsに-ObjC –all\_loadオプションを設定します。



* + - * 1. Build PhasesのLink Binary With Librariesに下記のライブラリを追加します。
  + MobileCoreServices.framework
  + SystemConfiguration.framework
  + MediaPlayer.framework
  + AVFoundation.framework
  + CoreMedia.framework



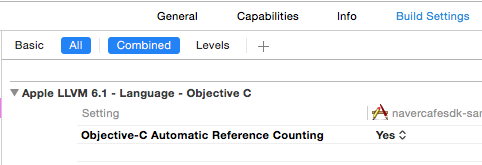
Cocos2d-xエンジンを使用する場合は、下記のライブラリを追加します。

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

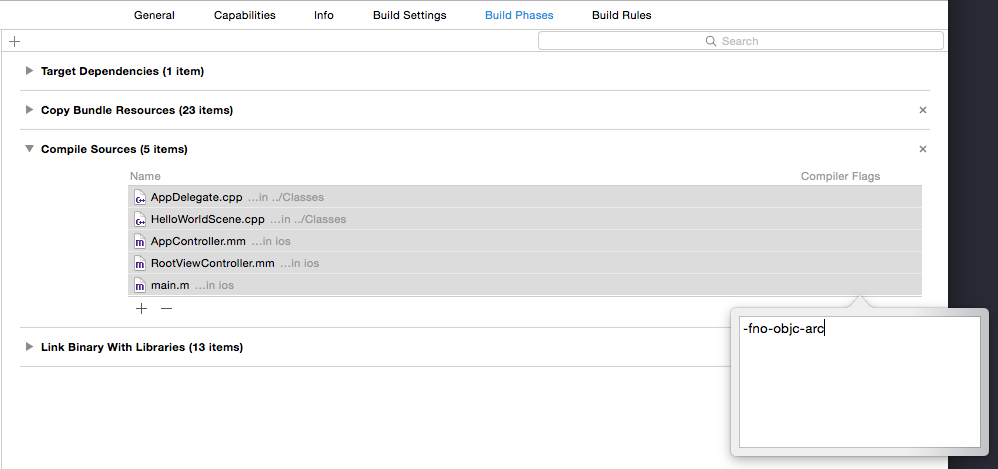
#### MRC環境をARC環境に切り替える

Cocos2d-xエンジンを使用するプロジェクトがMRC(manual reference counting)環境であれば、下記の方法でARC(automatic reference counting)環境に切り替えます。

* + - * 1. Build SettingsのObjective-C Automatic Reference CountingをYESに設定します。



* + - * 1. Build PhasesのCompile SourcesにてコンパイルするファイルのCompile Flagsを-fno-objc-arcに設定します。



## NAVER Cafe SDKの組み込み手順

### NAVER Cafe SDKの初期化

下記のようにNAVER Cafe SDKの初期化処理を行います。クライアントIDとクライアントシークレットは、NAVER Loginにアプリケーションを登録した際に発行された値です。

//ViewController

#import <NAVER CafeSDK/NCSDKManager.h>

- (void)viewDidLoad {

// NAVER Login登録情報とCafe IDでNAVER Cafe SDKの初期化処理を行います。

[[NCSDKManager getSharedInstance] setNaverLoginConsumerKey:@"クライアントID"

naverLoginConsumerSecret:@"クライアントシークレット"

cafeId:00000000];

}

### NAVER Loginの設定

下記のようにアプリデリゲートを設定し、NAVER Login処理が完了したらNAVER Cafe SDKにログイン情報を設定します。

//AppDelegate

#import <NAVER CafeSDK/NCSDKLoginManager.h>

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

//ログイン情報をNAVER Loginオブジェクトに設定します。

return [[NCSDKLoginManager getSharedInstance] finishNaverLoginWithURL:url];

}

### NAVER Cafe SDK画面を開く

NAVER Cafe SDK画面を立ち上げるにはNAVER Cafe SDKビューコントローラを設定し、 presentMainViewControllerメソッドを実行します。

#import <NAVER CafeSDK/NCSDKManager.h>

//NAVER Cafeトップを実行します。

[[NCSDKManager getSharedInstance] setParentViewController:self];

[[NCSDKManager getSharedInstance] presentMainViewController];

presentMainViewControllerメソッドの代わりにpresentMainViewControllerWithTabIndexメソッドを使えばNAVER Cafe SDK画面の特定のメニューを選択してNAVER Cafe SDK画面を立ち上げることができます。なお、presentMainViewControllerWithArticleIdメソッドを使用すると、特定掲示板の投稿を開いた状態でNAVER Cafe SDK画面を立ち上げることができます。

### 投稿作成

Cafe掲示板の投稿作成画面を開くには、下記のようにpresentArticlePostViewControllerWithMenuIdメソッドを使用します。

//投稿作成画面を実行します。

[[NCSDKManager getSharedInstance] setParentViewController:self];

[[NCSDKManager getSharedInstance] presentArticlePostViewControllerWithMenuId:0 subject:@"投稿のタイトル" content:@"投稿の内容" filePath:@"document"];

## APIリファレンス

### NCSDKManager

NAVER Cafe SDKのコントロールクラス。NCSDKManagerクラスのメソッドを使ってNAVER Cafe SDKを初期化したり、開始、中止する機能を実装します。NAVER Cafe SDKを実行する最上位のビューコントローラであるparentViewControllerクラスを持っています。

NCSDKManagerクラスのメソッドは、下記のとおりです。

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

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

#### 説明

NAVER Cafe SDKを実行するビューコントローラ。NAVER Cafe SDKを実行する親クラスです。

構文

@property (nonatomic, weak) id parentViewController;

パラメータ

なし

返却値

UIViewControllerオブジェクト

サンプルコード

[[NCSDKManager getSharedInstance] setParentViewController:self]

### (void)dismissViewController

#### 説明

NAVER Cafe SDKを実行するビューコントローラ上で実行された別のビューコントローラを消去します。

#### 構文

- (void)dismissViewController:(id)viewController;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| viewController | id | Y | NAVER Cafe SDKビューコントローラ上で実行されたビューコントローラ |

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] dismissViewController:self];

### (void)dismissTopViewController

#### 説明

NAVER Cafe SDKを実行するビューコントローラ上で実行された別のビューコントローラのうち、最上位にあるビューコントローラを消去します。

#### 構文

- (void)dismissTopViewController;

パラメータ

なし

返却値

なし

サンプルコード

[NCSDKManager getSharedInstance] dismissTopViewController];

### (NCSDKManager \*)getSharedInstance

#### 説明

NAVER Cafe SDKインスタンス(シングルトン・インスタンス)を獲得します。

#### 構文

+ (NCSDKManager \*)getSharedInstance;

パラメータ

なし

返却値

NCSDKManagerオブジェクト

サンプルコード

[NCSDKManager getSharedInstance]

### (id)NAVER CafeRootViewController

#### 説明

NAVER Cafe SDKの最上位ビューコントローラオブジェクトを獲得します。

#### 構文

- (id)NAVER CafeRootViewController;

パラメータ

なし

返却値

UIViewControllerオブジェクト

サンプルコード

[[NCSDKManager getSharedInstance] NAVER CafeRootViewController]

### (void)resetSharedInstance

#### 説明

NCSDKManagerオブジェクトを削除します。

#### 構文

+ (void)resetSharedInstance;

パラメータ

なし

返却値

なし

サンプルコード

[NCSDKManager resetSharedInstance]

### (void)presentArticlePostViewControllerWithMenuId

#### 説明

掲示板の投稿作成画面を開きます。

#### 構文

- (void)presentArticlePostViewControllerWithMenuId:(NSInteger)menuId

subject:(NSString \*)subject

content:(NSString \*)content;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| menuId | NSInteger | Y | 掲示板のID(基本値: 0)。掲示板のIDは、NAVER Cafe掲示板URLのmenuidパラメータ値にあたります。 |
| subject | NSString | N | 投稿のタイトル |
| content | NSString | N | 投稿の内容 |

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] presentArticlePostViewControllerWithMenuId:1

subject:@"タイトル"

content:@"内容"];

### (void)presentArticlePostViewControllerWithType

#### 説明

ファイルを添付して掲示板の投稿作成画面を開きます。

#### 構文

- (void)presentArticlePostViewControllerWithType:(GLArticlePostType)type

menuId:(NSInteger)menuId

subject:(NSString \*)subject

content:(NSString \*)content

filePath:(NSString \*)filePath;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| type | GLArticlePostType | Y | 添付ファイルのタイプ   * 1: 画像ファイル * 2: 動画ファイル |
| menuId | NSInteger | Y | 掲示板のID(基本値: 0)。掲示板のIDは、NAVER Cafe掲示板URLのmenuidパラメータ値にあたります。 |
| subject | NSString | N | 投稿のタイトル |
| content | NSString | N | 投稿の内容 |
| filePath | NSString | Y | 添付ファイルのパス |

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] presentArticlePostViewControllerWithType:kGLArticlePostTypeVideo

menuId:1

subject:@"タイトル"

content:@"内容"

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

### (void)presentMainViewController

#### 説明

NAVER Cafe SDK画面を開きます。

#### 構文

- (void)presentMainViewController;

パラメータ

なし

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] presentMainViewController];

### (void)presentMainViewControllerWithArticleId

#### 説明

掲示板の投稿記事が開いている状態でNAVER Cafe SDK画面を開きます。

#### 構文

- (void)presentMainViewControllerWithArticleId:(NSUInteger)articleId;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| articleId | NSUInteger | Y | 投稿ID(基本値: 0) |

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] presentMainViewControllerWithArticleId:10];

### (void)presentMainViewControllerWithTabIndex

#### 説明

指定したメニューが選択された状態でNAVER Cafe SDK画面を開きます。

#### 構文

- (void)presentMainViewControllerWithTabIndex:(NSUInteger)tabIndex;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| tabIndex | NSUInteger | Y | 選択するメニューのインデックス値(基本値: 0)   * 0: トップ * 1: お知らせ * 2: イベント * 3: 統合掲示板 * 4: プロフィール |

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] presentMainViewControllerWithTabIndex:1];

### (void)presentViewController

#### 説明

NAVER Cafe SDKの最上位ビューコンローラから他のビューコントローラを実行します。

#### 構文

- (void)presentViewController:(id)viewController;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| viewController | id | Y | ビューコントローラオブジェクトのID |

返却値

なし

サンプルコード

[[NCSDKManager getSharedInstance] presentViewController:self];

### (void)setGameUserId

#### 説明

ユーザーのゲームIDとNAVER IDを連携してプロフィール画面にIDを表示します。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| gameUserId | NSString | Y | ユーザーのゲームID |
| fieldName | NSString | N | プロフィール画面に表示するID(基本値: ゲームID) |

返却値

なし

サンプルコード

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

### (void)setNaverLoginClientId

#### 説明

NAVER Loginオブジェクトを設定します。

#### 構文

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

naverLoginClientSecret:(NSString \*)naverLoginClientSecret

cafeId:(NSInteger)cafeId;

パラメータ

なし

返却値

なし

サンプルコード

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

naverLoginConsumerSecret:@"Secret ID"

cafeId:00000000];

### NCSDKLoginManager

NAVER Cafe SDKでNAVER Login機能をコントロールするクラス。

NCSDKLoginManagerクラスのメソッドは下記のとおりです。

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

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

#### 説明

ログインのウェブビューを実行するビューコントローラ。

#### 構文

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

パラメータ

なし

返却値

UIViewControllerオブジェクト

サンプルコード

[[NCSDKLoginManager getSharedInstance] setRootViewController:self];

### (NSString \*)accessToken

#### 説明

NAVER　Login認証後にNAVERサーバから受け取ったアクセストークン(access token)が返されます。

#### 構文

- (NSString \*)accessToken;

パラメータ

なし

返却値

アクセストークン

サンプルコード

[[NCSDKLoginManager getSharedInstance] accessToken];

### (NSString \*)accessTokenExpireTime

#### 説明

アクセストークン(access token)の満了期限を確認します。

#### 構文

- (NSString \*)accessTokenExpireTime;

パラメータ

なし

返却値

アクセストークン満了期限

サンプルコード

[[NCSDKLoginManager getSharedInstance] accessTokenExpireTime];

### (BOOL)finishNaverLoginWithURL

#### 説明

NAVER IDでのログイン後にアプリデリゲートを実行します。

#### 構文

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

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| url | NSURL | Y | NAVER IDでのログイン後にアプリデリゲートでコールバックされるURL Scheme |

返却値

* true: ログインに成功
* false: ログインに失敗

サンプルコード

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

return [[NCSDKLoginManager getSharedInstance] finishNaverLoginWithURL:url];

}

### (NCSDKLoginManager \*)getSharedInstance

#### 説明

NAVER Loginインスタンス(シングルトン・インスタンス)を獲得します。

#### 構文

+ (NCSDKLoginManager \*)getSharedInstance;

パラメータ

なし

返却値

NCSDKLoginManagerオブジェクト

サンプルコード

[NCSDKLoginManager getSharedInstance]

### (void)isLoginWithFinish

#### 説明

NAVER　Login認証後にNAVERサーバから受け取ったアクセストークン(access token)の有効有無を確認します。

#### 構文

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

パラメータ

なし

返却値

* true: 有効なアクセストークンです。
* false: アクセストークンの期限が満了になり、無効になったアクセストークンです。

サンプルコード

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

if (successACToken) {

}

}];

### (BOOL)isValidAccessTokenExpireTimeNow

#### 説明

アクセストークン(access token)があるか、有効期限は残っているかを確認します。ただし、サーバーで有効期限が満了した場合には確認できません。

#### 構文

- (BOOL)isValidAccessTokenExpireTimeNow;

パラメータ

なし

返却値

* true: 有効なアクセストークンです。
* false: アクセストークンの期限が満了になり、無効になったアクセストークンです。

サンプルコード

[[NCSDKLoginManager getSharedInstance] isValidAccessTokenExpireTimeNow];

### (void)loginWithFinish

#### 説明

NAVER Login認証を開始し、アクセストークン(access token)が返されます。

#### 構文

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

パラメータ

なし

返却値

* true: 有効なアクセストークンです。
* false: アクセストークンの期限が満了になり、無効になったアクセストークンです。

サンプルコード

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

if (successACToken) {

}

}];

### (void)logout

#### 説明

クライアントに保存されているアクセストークン(access token)と更新トークン(refresh token)を削除してログアウトします。

#### 構文

- (void)logout;//delete local accesstoken

パラメータ

なし

返却値

なし

サンプルコード

[[NCSDKLoginManager getSharedInstance] logout];

### (void)refreshAccessToken

#### 説明

NAVER　Login認証後にNAVERサーバーから受け取った更新トークン(refresh token)でアクセストークン(access token)を更新します。

#### 構文

- (void)refreshAccessToken;

パラメータ

なし

返却値

なし

サンプルコード

[[NCSDKLoginManager getSharedInstance]refreshAccessToken];

### (void)refreshAccessTokenWithFinish

#### 説明

NAVER Login認証後にNAVERサーバーから更新トークン(refresh token)が返されます。

#### 構文

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

パラメータ

なし

返却値

* true: アクセストークンの更新に成功
* false: アクセストークンの更新に失敗

サンプルコード

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

if (successACToken) {

}

}];

### (void)requestDeleteToken

#### 説明

クライアントとサーバーに保存されているアクセストークン(access token)と更新トークン(referesh token)を削除します。

#### 構文

- (void)requestDeleteToken;//delete server authorization

パラメータ

なし

返却値

なし

サンプルコード

[[NCSDKLoginManager getSharedInstance] requestDeleteToken];

### (void)setIsInAppOauthEnable

#### 説明

アプリ内ブラウザを実行してログイン手続きを行います。

#### 構文

- (void)setIsInAppOauthEnable:(BOOL)enable;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| enable | BOOL | Y | アプリ内ブラウザの実行有無 |

返却値

なし

サンプルコード

[[NCSDKLoginManager getSharedInstance] setIsInAppOauthEnable:YES];

### (void)setIsNaverAppOauthEnable

#### 説明

NAVERアプリを実行してログイン手続きを行います。

#### 構文

- (void)setIsNaverAppOauthEnable:(BOOL)enable;

パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| enable | BOOL | Y | NAVERアプリの実行有無 |

返却値

なし

サンプルコード

[[NCSDKLoginManager getSharedInstance] setIsNaverAppOauthEnable:YES];

# Unity向け

## 開発環境

### ソフトウェア要件

#### 開発ツール

* ゲームエンジン: Unity 4以上

### ライブラリの構成

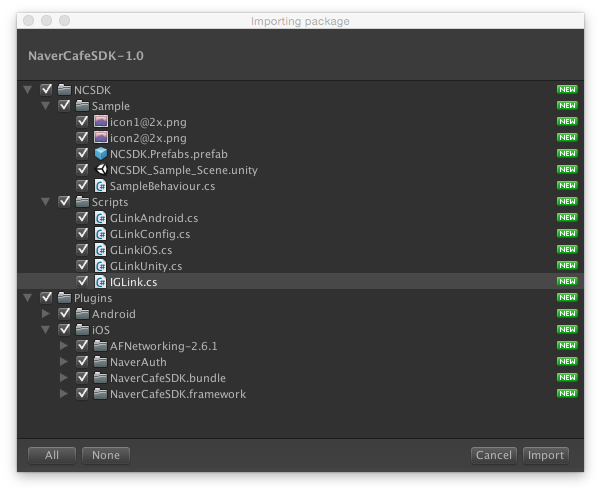
* NAVER CafeSDK-1.0.unitypackage: Unity向けNAVER Cafe SDKライブラリ

### 開発環境セットアップ

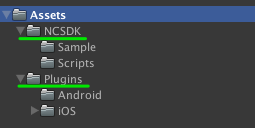
NAVER Cafe SDKを組み込むには下記のように開発環境をセットアップします。

#### Unityの設定

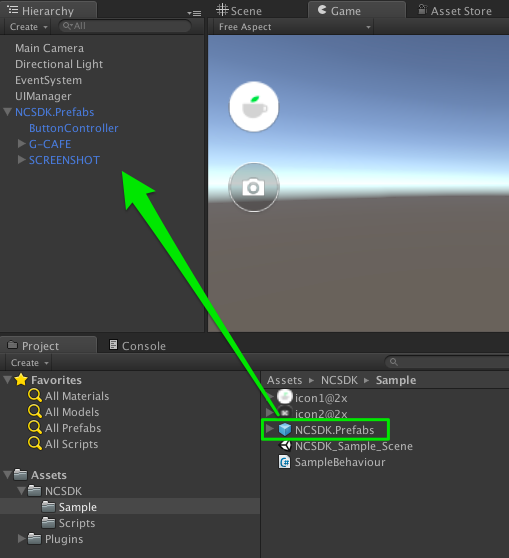
* + - * 1. Unityメニューから、Asset > Import packageをクリックしCustom PackageダイアログボックスからダウンロードしたNAVER CafeSDK-1.0.unityPackageファイルを選択します。



* + - * 1. Custom PackageダイアログボックスからImportをクリックするとNCSDKフォルダとPluginsフォルダが作成されます。



* + - * 1. NCSDKフォルダのSampleフォルダにあるNCSDK.Prefabsを希望のシーンへドラッグします。NAVER Cafe SDKで基本提供するボタンが作成されます。



* + - * 1. NCSDKフォルダのScriptsフォルダにあるGLinkConfigファイルにCafe IDとクライアントID、クライアントシークレットを入力します。

//NCSDK/Scripts/GLinkConfig

static class GLinkConfig

{

public const string NaverLoginConsumerKey =

"Consumer ID";

public const string NaverLoginConsumerSecret =

"Secret ID";

public const int CafeId =

00000000;

}

#### Androidの設定

Unity 5エンジンを使用する場合は、下記のようにNAVER Cafe SDKライブラリを追加します。

* NAVER Cafe SDKライブラリファイル(cafeSdk-x.x.x.aarファイル)と必須ライブラリファイルを Assets/Plugins/Androidフォルダに追加します。

Unity 4エンジンを使用する場合は、Eclipseで下記の追加設定が必要です。

* + - * 1. NAVER Cafe SDKライブラリファイルのうち、cafeSdk-x.x.x.zipファイルを解凍します。
        2. 解凍したNAVER Cafe SDK ライブラリフォルダ内のlibsフォルダにビルドに必要な必須ライブラリファイルを追加します。
        3. Eclipseから解凍したフォルダをプロジェクトで読み込みます。
        4. プロジェクト設定ダイアログボックスからAndroidをクリックしてIs Libraryを選択します。AndroidManifest.xmlファイルにて、下記のようにNAVER Cafe SDKで使用するアクティビティを追加します。

<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"

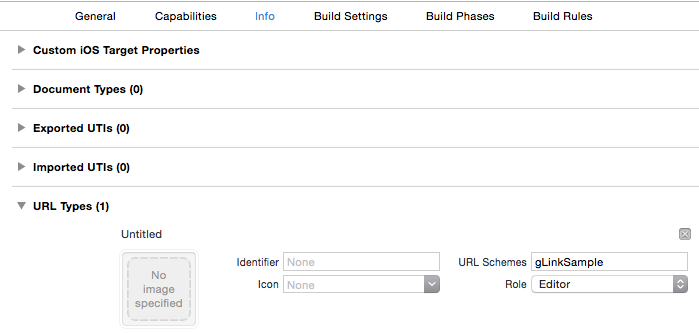
android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"/>

* + - * 1. プロジェクトをビルドします。
        2. EclipseでビルドしたファイルをUnityプロジェクトのAssets/Plugins/Androidフォルダに追加します。
        3. Unityプロジェクトをビルドします。

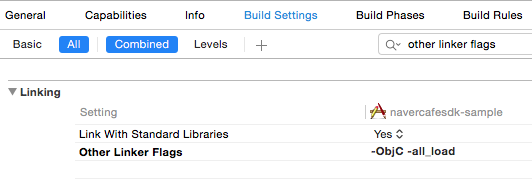
#### iOSの設定

iOSアプリケーションを開発する際は、下記のようにXcodeを設定します。

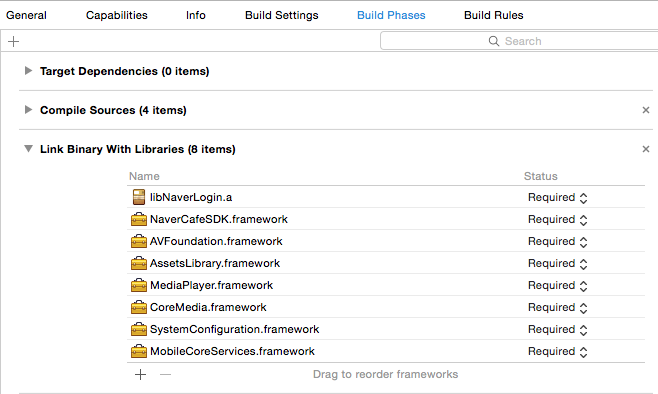
* + - * 1. NAVER Loginでアプリケーション登録時に入力したURL SchemeをXcodeプロジェクトに登録します。



* + - * 1. スタティックライブラリを使えるようにBuild SettingsのOther Linker Flagsに-ObjC –all\_loadオプションを設定します。



* + - * 1. Build PhasesのLink Binary With Librariesに下記のライブラリを追加します。
  + MobileCoreServices.framework
  + SystemConfiguration.framework
  + MediaPlayer.framework
  + AVFoundation.framework
  + CoreMedia.framework
  + Security.framework
  + AssetsLibrary.framework



* + - * 1. 下記のようにアプリデリゲートを設定し、NAVER　Loginが完了したらアプリデリゲートを呼び出します。

//AppDelegate

# import <NAVER CafeSDK/NCSDKLoginManager.h>

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

return [[NCSDKLoginManager getSharedInstance] finishNaverLoginWithURL:url];

}

## NAVER Cafe SDKの組み込み手順

### 実行

NAVER Cafe SDK画面を立ち上げるには、下記のようにGlinkUnity.executeMain()メソッドを呼び出します。

// Cafeトップ

GlinkUnity.executeMain ();

投稿作成画面は、下記のようにGlinkUnity.executeArticlePostWithImgae()メソッドで実装します。

//スクリーンショット

GlinkUnity.executeArticlePostWithImgae(menuId, "投稿のタイトル", "投稿の内容", image path);

GlinkUnity .executeArticlePostWithImage()メソッドやGlinkUnity .executeArticlePostWithVideo() メソッドを使用すると画像や動画を添付した状態で掲示板の投稿作成画面を開くことができます。

### ユーザーのゲームIDとNAVER IDとのマッピング

NAVER Cafe SDKは、ユーザーのゲームIDとNAVER IDをマッピングする機能を提供します。PCのCafe管理者ページへアクセスすると、マッピングされたゲームIDとNAVER IDのリストを確認することができます。

下記は、iOSアプリケーションでゲームIDとNAVER IDをマッピングする機能を実装した例です。

/\*

NAVER IDとゲームIDをマッピングさせるためにユーザーのゲームIDを連携します。

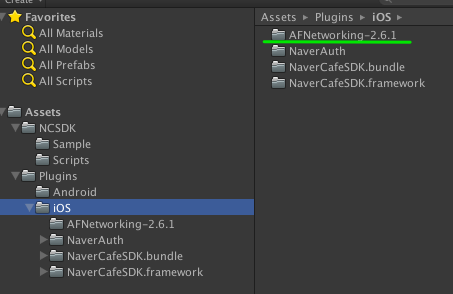
fieldNameは、単純にプロフィールエリアに表示する文字列です(基本値: ゲームID)。

\*/

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

### AFNetworkingの交替

NAVER CafeSDK.unitypackageには、ネットワーク通信のためのライブラリであるAFNetworking 2.6.1が基本的に含まれています。プロジェクトの状況によってAFNetworkingライブラリを削除したり、別のバージョンでビルドしても構いません。ただし、NAVER Cafe SDKはAFNetworking 1.x以上に対応します。



## APIリファレンス

### GLinkUnity

NAVER Cafe SDKのコントロールクラス。GLinkUnityクラスのメソッドを使ってNAVER Cafe SDKを開始したり、投稿作成画面を開く機能を実装します。

GLinkUnityクラスのメソッドは、下記のとおりです。

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

### executeArticlePost()

#### 説明

掲示板の投稿作成画面を立ち上げます。

#### 構文

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

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

}

#### パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| menuId | int | Y | 掲示板のID(基本値: -1). 掲示板のIDは、NAVER Cafe掲示板のURLのmenuidパラメータ値にあたります。 |
| subject | string | Y | 投稿のタイトル |
| content | string | Y | 投稿の内容 |

#### 返却値

なし

#### サンプルコード

GlinkUnity.executeArticlePost(28290504, "投稿のタイトル","投稿の内容" );

### executeArticlePostWithImage()

#### 説明

画像を添付した状態で掲示板の投稿作成画面を立ち上げます。

#### 構文

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

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

}

#### パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| menuId | int | Y | 掲示板のID(基本値: -1). 掲示板のIDは、NAVER Cafe掲示板URLのmenuidパラメータ値にあたります。 |
| subject | string | Y | 投稿のタイトル |
| content | string | Y | 投稿の内容 |
| filePath | string | Y | 画像ファイルのパス(URI形式) |

#### 返却値

なし

#### サンプルコード

{

GlinkUnity.executeArticlePostWithImgae(28290504, "投稿のタイトル", "投稿の内容","/NAVER Cafesdk/glink.png" );

}

### executeArticlePostWithVideo()

#### 説明

動画を添付した状態で掲示板の投稿作成画面を立ち上げます。

#### 構文

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

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

}

#### パラメータ

|  |  |  |  |
| --- | --- | --- | --- |
| パラメータ | タイプ | 必須 | 説明 |
| menuId | int | Y | 掲示板のID(基本値: -1). 掲示板のIDは、NAVER Cafe掲示板のURLのmenuidパラメータ値にあたります。 |
| subject | string | Y | 投稿のタイトル |
| content | string | Y | 投稿の内容 |
| filePath | string | Y | 動画ファイルのパス(URI形式) |

#### 返却値

なし

#### サンプルコード

{

GlinkUnity.executeArticlePostWithImgae(28290504, "投稿のタイトル", "投稿の内容","/NAVER Cafesdk/glink.avi" );

}

### executeMain()

#### 説明

NAVER Cafe SDK画面を立ち上げます。

#### 構文

public static void executeMain() {

sharedInstance().executeMain ();

}

#### パラメータ

なし

#### 返却値

なし

#### サンプルコード

{

GlinkUnity.executeMain ();

}