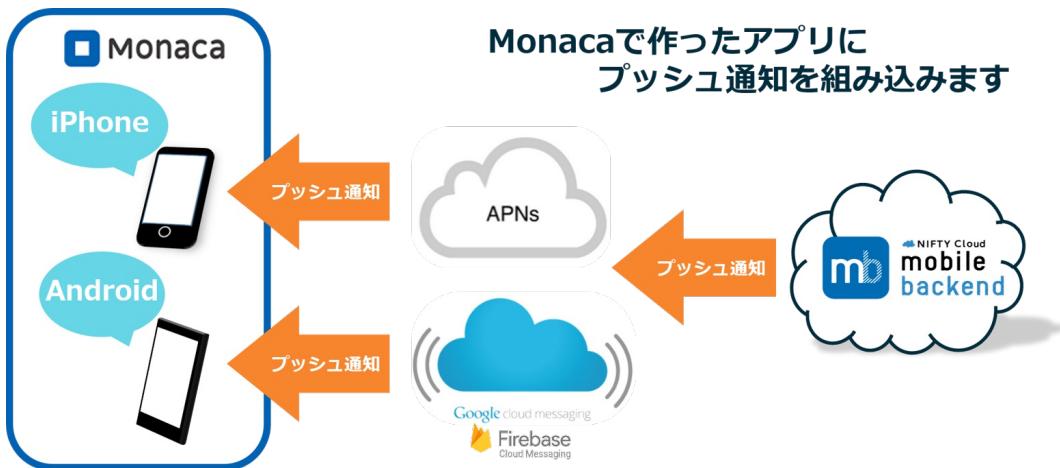


# 【Monaca】 アプリにプッシュ通知を組み込もう！



## 概要

- ニフティクラウドmobile backendの『プッシュ通知』機能を実装したサンプルプロジェクトです
- 簡単な操作ですぐにニフティクラウドmobile backendの機能を体験いただけます★☆

## ニフティクラウドmobile backendとは

スマートフォンアプリのバックエンド機能（プッシュ通知・データストア・会員管理・ファイルストア・SNS連携・位置情報検索・スクリプト）が開発不要、しかも基本無料(注1)で使えるクラウドサービス！



注1：詳しくは[こちら](#)をご覧ください

# 動作環境の準備

---

## 共通

- Monaca 会員登録
  - 下記リンクより登録（無料）をお願いします  
<https://ja.monaca.io/>
- ニフティクラウド mobile backend 会員登録
  - 下記リンクより登録（無料）をお願いします  
<http://mb.cloud.nifty.com/>
- 推奨ブラウザ環境の準備
  - Chrome最新版

## Android端末で動作確認をする場合

- PC
- Googleアカウント
- Android端末（最新バージョン推奨）

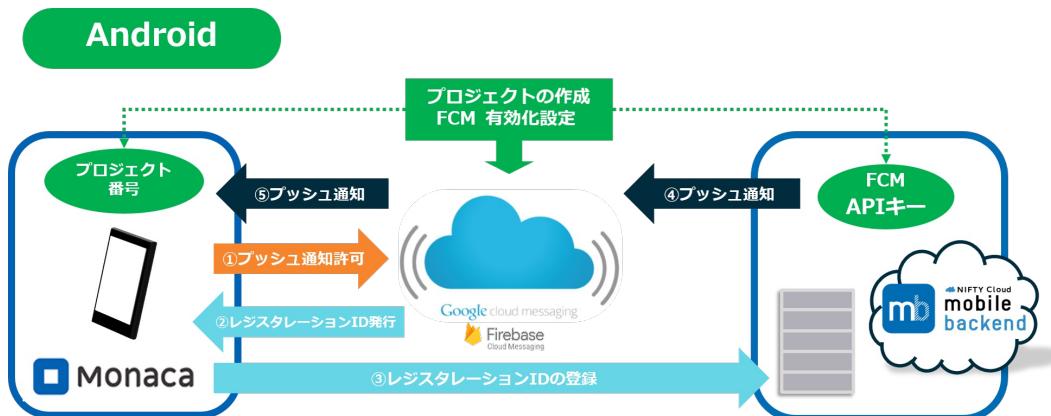
## iOS端末で動作確認をする場合

- Mac（キーチェーンアクセスを利用）
- Apple Developer Program (有償)アカウント
  - 別の Mac で使用しているアカウントの場合、発行する証明書に秘密鍵を紐付けることができません。ただし、アカウントを使用している Mac から秘密鍵を書き出して、今回使用するMacに送ることで作業は可能です
- iOS 端末（最新バージョン推奨）
- Lightning ケーブル（端末の UDID を調べるために必要です）

# プッシュ通知の仕組み

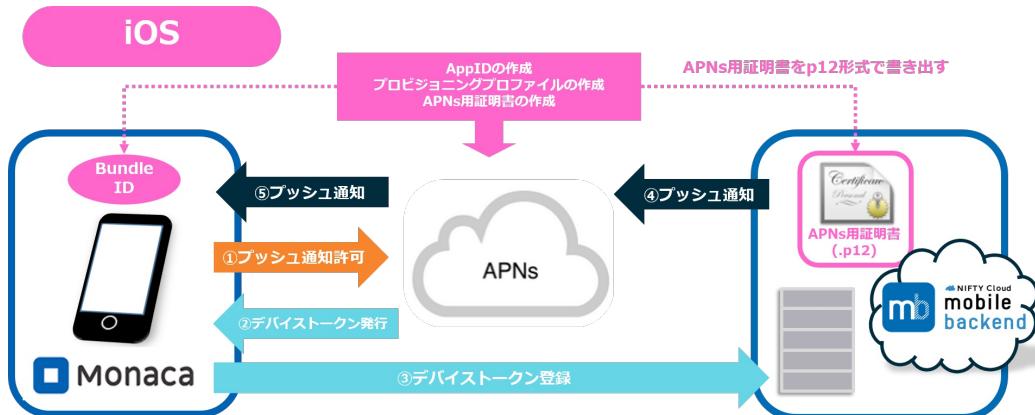
ニフティクラウド mobile backend のプッシュ通知は、各プラットフォームが提供している通知サービスを利用しています。

## Androidの通知サービス FCM (Firebase Cloud Messaging)



- FCM は GCM (Google Cloud Messaging)の新バージョンです。既に GCM にてプロジェクトの作成・GCM の有効化設定を終えている場合は、継続してご利用いただくことが可能です。新規で GCM をご利用いただくことはできませんので、あらかじめご了承ください。

## iOSの通知サービス APNs (Apple Push Notification Service)



- 上図のように、アプリ（Monaca）・サーバー（ニフティクラウド mobile backend）・通知サービス（FCMあるいはAPNs）の間で認証が必要になります
  - 認証に必要な鍵や証明書の作成は作業手順の「0. プッシュ通知機能を使うための準備」で行います

# 作業の手順

## 0. プッシュ通知機能を使うための準備

動作確認を行う端末に応じて該当する内容を準備してください

### Android端末で動作確認をする場合

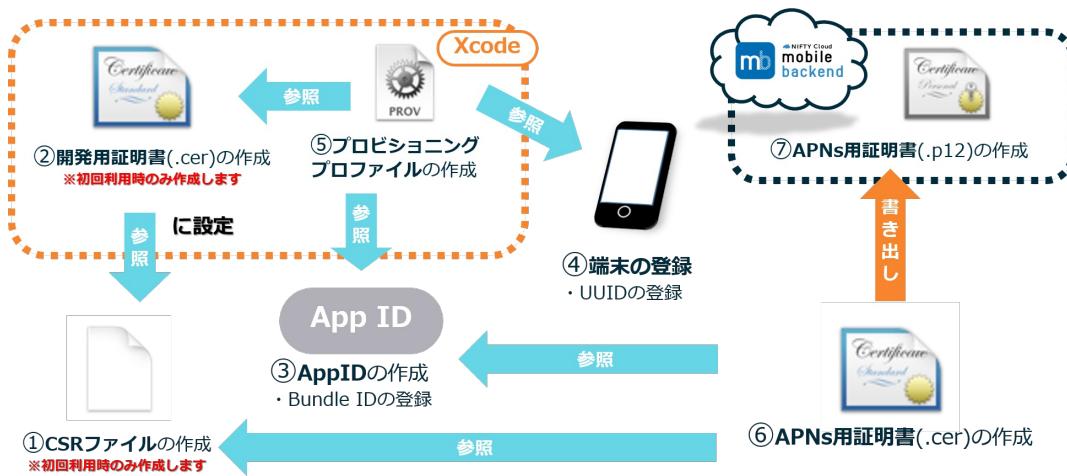
#### Android 端末で動作確認されるかたはこちら

- ニフティクラウド mobile backend と連携させるための APIキー(サーバーキー)と端末情報の登録処理時に必要な Sender ID (送信者ID) を取得する必要があります
- 下記リンク先のドキュメントを参考に、FCM プロジェクトの作成と APIキー・Sender IDの取得を行ってください

### iOS端末で動作確認をする場合

#### iOS 端末で動作確認されるかたはこちら

- 上記のドキュメントをご覧の上、必要な証明書類の作成をお願いします
- 証明書の作成には[Apple Developer Program](#)の登録（有料）が必要です



# 1. ニフティクラウド mobile backend の準備

- ニフティクラウド mobile backend にログインします  
<http://mb.cloud.nifty.com/>



- 新しいアプリを作成します
- アプリ名を入力し、「新規作成」をクリックします
  - 例) PushDemo



- mobile backend を既に使用したことがある場合は、画面上方のメニューにある「+新しいアプリ」をクリックすると同じ画面が表示されます



- アプリ作成されると下図のような画面になります
- この2種類のAPIキー（アプリケーションキーとクライアントキー）は、この後 Monaca で作成するアプリとの連携のために使用します



- 続けて、「0. プッシュ通知機能を使うための準備」で動作確認端末別に作成した認証キーまたは証明書を設定します



### 共通

**プッシュ通知**  
**プッシュ通知の許可**  
 →「許可する」を選択して「保存する」をクリック

### Android

**Androidプッシュ通知**  
**APIキー**  
 →FCMのAPIキーを入力して「保存する」をクリック

### iOS

**iOSプッシュ通知**  
**証明書 (.p12)**  
 →「証明書の選択」をクリックして証明書をアップロード

- mobile backend 側の準備は以上です

## 2. Monaca でアプリの準備

- Monacaにログインをします

<https://ja.monaca.io/>



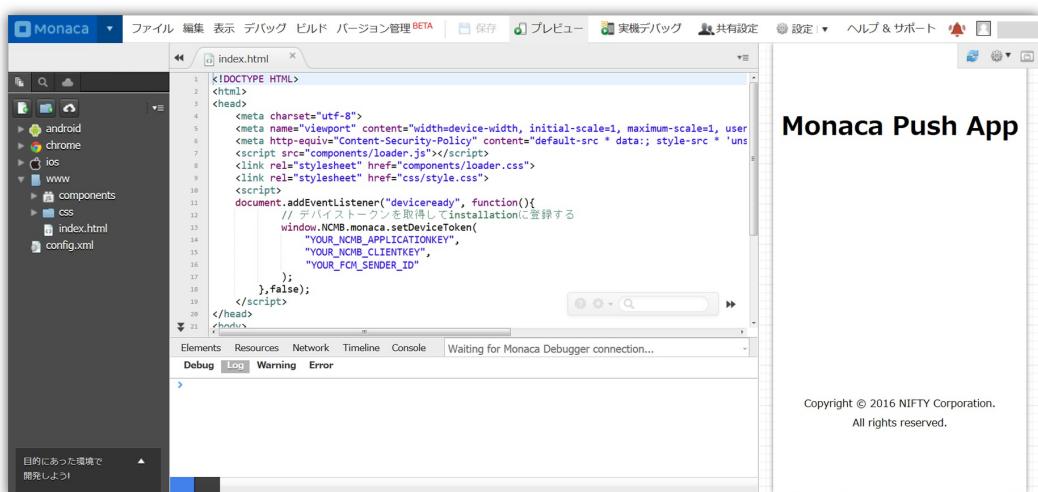
- プロジェクトを作成します



- プロジェクト名を入力します
- 説明は空欄でOKです
- インポート方法は「URLを指定してインポート」を選択し、下記リンクを右クリックしてURLをコピーしたものを貼り付けてください
  - <https://github.com/NIFTYCloud-mbaas/MonacaPushApp/archive/master.zip>

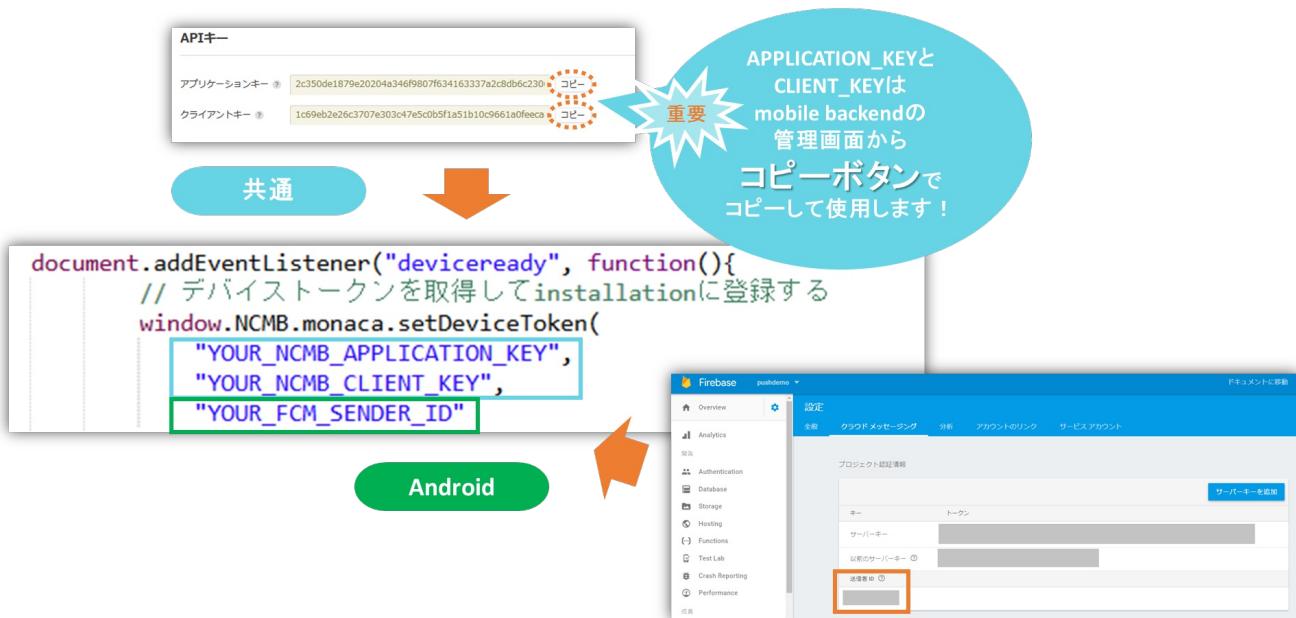


- 作成されたプロジェクトを「開く」をクリックして開きます



### 3. APIキーの設定

- 作成した Monaca プロジェクトに mobile backend で発行した API キーを設定して連携します
- www/index.html を編集します
  - Monacaでプロジェクトを開くとデフォルトで表示されるファイルです
- 中ほどにある、YOUR\_NCMB\_APPLICATION\_KEY と YOUR\_NCMB\_CLIENT\_KEY の部分を、先程 mobile backend のダッシュボード上で確認した APIキーに書き換えます
- Android端末で動作確認をする場合**は、YOUR\_FCM\_SENDER\_ID を FCM で発行した Sender ID (送信者ID)に書き換えます
  - iOS端末で動作確認をする場合は、そのままOKです



- このとき、ダブルクオーテーション ("") を消さないように注意してください！

- 書き換え終わったら「保存」をクリックして保存します
  - メニューバーの「保存」をクリックします
  - Windowsの場合は、「Ctrl + S」でも保存可能
  - Macの場合は、「Command + S」でも保存可能

こんな感じになつていればOKです[イメージ]



The screenshot shows the Monaca IDE interface. The top menu bar includes 'Monaca' (with a dropdown), 'ファイル' (File), '編集' (Edit), '表示' (View), 'デバッガ' (Debugger), 'ビルト' (Build), 'バージョン管理' (Version Management) BETA, '保存' (Save) with a blue icon, 'プレビュー' (Preview), '実機デバッガ' (Physical Device Debugger), '共有設定' (Shared Settings), and '設定' (Settings). Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main workspace shows code for 'index.html' in a monospaced font. The code includes meta tags for viewport and security, links to components and loader scripts, and a script block containing device token logic. A sidebar on the left lists project files: 'README.md', 'index.html', and 'config.xml'. The status bar at the bottom shows the file path 'niftycloud mobile b...' and line numbers 1 through 28.

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
6   <meta http-equiv="Content-Security-Policy" content="default-src * data: style-src * 'unsafe-inline'; script-src * 'unsafe-inlini...
7   <link rel="stylesheet" href="components/loader.css">
8   <link rel="stylesheet" href="css/style.css">
9   <script src="components/loader.js"></script>
10  <script src="js/ncmb-2.1.1.min.js"></script>
11  <script>
12    // Add your code here
13    document.addEventListener("deviceready", function(){
14      // デバイストークンを取得してinstallationに登録する
15      window.NCMB.monaca.setDeviceToken(
16        "a00925ffac8d9ce7c[REDACTED]379f8a67da61237ce6636c064102d2",
17        "4fce697bb07d506f0[REDACTED]2889e36cb9eff9841fd4b31b0fc4e1",
18        "4977[REDACTED]340"
19      );
20    },false);
21  </script>
22 </head>
23 <body>
24
25   <p>App template using NIFTY Cloud mobile backend</p>
26
27 </body>
28 </html>

```

- これで Monaca プロジェクトと mobile backend が連携されました！準備は完了です◎

## 4. 実機ビルド

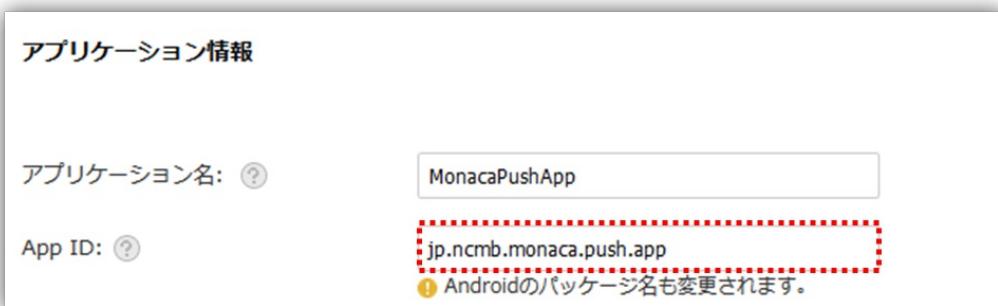
- 動作確認を行う端末に応じて、該当する作業を行ってください

### Android端末で動作確認をする場合

- 「ビルド」 > 「Androidアプリのビルド」をクリックします
- デフォルト設定（「デバッグ向けビルド」 > 「デバッグビルド」が選択されている状態）で画面下の「ビルドを開始する」ボタンをクリックします
- 別のタブでビルドが開始され、少し待つとビルドが完了します
- ビルド完了後の画面に表示されるいずれかの方法で、端末にアプリをインストールをしてください

### iOS端末で動作確認をする場合

- まず「設定」 > 「iOSアプリ設定...」をクリックします
- App ID を変更します
  - App ID 作成時に設定した『Bundle ID』に書き換えます



- 画面下の「保存する」ボタンをクリックします
- 次に「設定」 > 「iOSビルド設定...」をクリックします
- 「秘密鍵と証明書のインポート」の「インポート」ボタンをクリックして、作成した『開発用証明書(秘密鍵.p12)』を設定します

**発行された証明書を登録する**

証明書とプロファイル	1. Apple Developer Programで生成した証明書 2. 証明書に紐づくプロファイルの順でアップロードしてください。	証明書のアップロード
秘密鍵と証明書のインポート	秘密鍵と証明書のインポート	プロファイルのアップロード

**秘密鍵のインポート**

Key Chain Accessツールからエクスポートした秘密鍵をインポートできます。詳細はマニュアルを参照してください。

秘密鍵（PKCS#12 .12）:  参照する  
キーのパスワード:

**「インポート」をクリック**

「参照する」をクリックして秘密鍵.p12を選択、パスワードは何も入力せず「インポート」をクリック  
※パスワードが設定されている場合は入力する

秘密鍵.p12

**Monacaに登録された証明書**

証明書	ステータス
Dev iPhone Developer: [REDACTED]	プロジェクトプロファイルが登録されていません。

証明書が表示されればOK！

- 「証明書とプロファイル」の「プロファイルのアップロード」ボタンをクリックして、作成した『プロジェクトプロファイル』を設定します

**発行された証明書を登録する**

証明書とプロファイル	1. Apple Developer Programで生成した証明書 2. 証明書に紐づくプロファイルの順でアップロードしてください。	証明書のアップロード
秘密鍵と証明書のインポート	秘密鍵と証明書のインポート	プロファイルのアップロード

**プロジェクトプロファイルを設定**

「プロファイルのアップロード」をクリックして、作成した『プロジェクトプロファイル』を設定

**Monacaに登録された証明書**

証明書	プロファイル
Dev iPhone Developer: [REDACTED]	Dev [REDACTED]

表示されればOK！

- 「ビルド」 > 「iOSアプリのビルド」をクリックします
- デフォルト設定（「デバッグ向けビルド」 > 「デバッグビルド」が選択されている状態）で画面下の「ビルドを開始する」ボタンをクリックします
- 別のタブでビルドが開始され、少し待つとビルドが完了します
- 下記のいずれかの方法で、端末にアプリをインストールをしてください

## iOSアプリのインストール方法

1. iTunesを使う方法
  - ダウンロードしたプロジェクト.ipaをドラッグ&ドロップ
2. Xcodeを使う方法
  - [http://docs.monaca.mobi/cur/ja/manual/deploy/non\\_market\\_deploy/](http://docs.monaca.mobi/cur/ja/manual/deploy/non_market_deploy/)
3. DeployGateを使う方法
  - アカウント（無料）を取得し、ダウンロードしたプロジェクト.ipaをドラッグ&ドロップ
  - <https://deploygate.com/>

## 5. 動作確認

- 端末でインストールしたアプリを起動します
  - プッシュ通知の許可を求めるアラートが出たら、必ず許可してください（iOS端末のみ）
- 起動されたら時点で Android端末は レジスタレーションID が、iOS 端末は デバイストークン が取得され、mobile backend に保存されます
- mobile backend のダッシュボードで「データストア」> 「installation」 クラスを確認してみましょう！

The screenshot shows the NIFTY Cloud mobile backend dashboard. On the left, there's a sidebar with icons for 'mobile backend', 'アプリ一覧' (App List), 'ダッシュボード' (Dashboard), 'ドキュメント' (Documentation), '開発TIPS' (Development TIPS), 'コミュニティ' (Community), and '連携サービス' (Connected Services). The main area is titled 'データストア' (Data Store) and shows a table for the 'installation' class. The table has columns: objectId, sdkVersion, badge, channels, deviceToken, deviceType, and appVersion. There are two rows of data. A blue callout bubble points to the 'deviceToken' column, containing values like 'dMyIDPi6rD0:APA91bGQ5Oo24ipzZEGHLjiCn...' and 'fOecuCdQm0:APA91bFN7Qalnk10eWrC4IN...'. A green callout bubble points to the top right corner of the screen, containing the text 'ここに入りました'.

	objectId	sdkVersion	badge	channels	deviceToken	deviceType	appVersion
□	QN1bGIMvdAeRFDIG	2.2.3	0	[]	dMyIDPi6rD0:APA91bGQ5Oo24ipzZEGHLjiCn...	android	1.0
□	cXjeL4nie49UV9Xk	2.2.0	0	[]	fOecuCdQm0:APA91bFN7Qalnk10eWrC4IN...	android	1.0

ここに登録され端末にプッシュ通知を送信することができます

ここに入りました

- 端末側で起動したアプリは一度閉じておきます

# 6. プッシュ通知を送りましょう！

- 実際にプッシュ通知を送ってみましょう！
- mobile backend のダッシュボードで「プッシュ通知」>「+新しいプッシュ通知」をクリックします
- プッシュ通知のフォームが開かれます
- 必要な項目を入力してプッシュ通知を作成します



- 端末を確認しましょう！
- 少し待つとプッシュ通知が届きます！！！

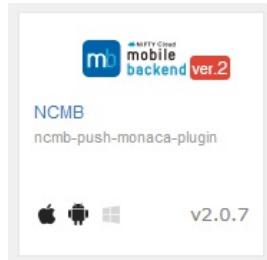


## 解説

サンプルプロジェクトに実装済みの内容のご紹介

### Cordovaプラグインの設定

- Monacaで「設定」>「Cordovaプラグインの管理...」を開きます
- プッシュ通知をアプリに実装する場合は以下のプラグインを有効にします



## ロジック

- index.html の `<script></script>` タグ内にデバイストークンを取得し、ニフティクラウドmobile backendに保存するロジックを書いています

```
document.addEventListener("deviceready", function(){
    // デバイストークンを取得してinstallationに登録する
    window.NCMB.monaca.setDeviceToken(
        "YOUR_NCMB_APPLICATION_KEY",
        "YOUR_NCMB_CLIENT_KEY",
        "YOUR_FCM_SENDER_ID"
    );
}, false);
```

- 「YOUR\_NCMB\_APPLICATION\_KEY」、「YOUR\_NCMB\_CLIENT\_KEY」はmobile backendのダッシュボードのアプリケーションキー、クライアントキーにそれぞれ書き換えてください
- Android端末で動作確認を行う場合は、「YOUR\_FCM\_SENDER\_ID」をFCMでプロジェクト作成時に発行されたSenderID（送信者ID）に書き換えてください

## 参考

- ニフティクラウドmobile backend のドキュメントもご活用ください
  - クイックスタート
  - プッシュ通知