

Monaca クイックスタート

このページでは、mobile backendをMonacaアプリと連携させる手順を紹介します

- [Monaca](#)とは、HTML5とJavaScriptでハイブリッドアプリがクラウド上で開発できる統合開発環境です。
- Monacaクラウドでの開発には下記のブラウザ環境が必要です
 - Google Chrome 最新版
- JavaScript、Adobe Flash が利用可能である必要があります

目次

- アプリの新規作成
- SDKのインストールと読み込み
- APIキーの設定とSDKの初期化
- サンプルコードの実装
 - サンプルコード（データストア）
 - アプリを実行してmBaaSのダッシュボードを確認する

アプリの新規作成

mBaaS ダッシュボード

- ニフティクラウドmobile backendにログインします
- ダッシュボードが表示されたら、「アプリの新規作成」を行います
- すでに別のアプリを作成済みの場合は、ヘッダーの「+新しいアプリ」をクリックします



- 「アプリ名」を入力し「新規作成」をクリックすると、APIキー（アプリケーションキーとクライアントキー）が発行されます

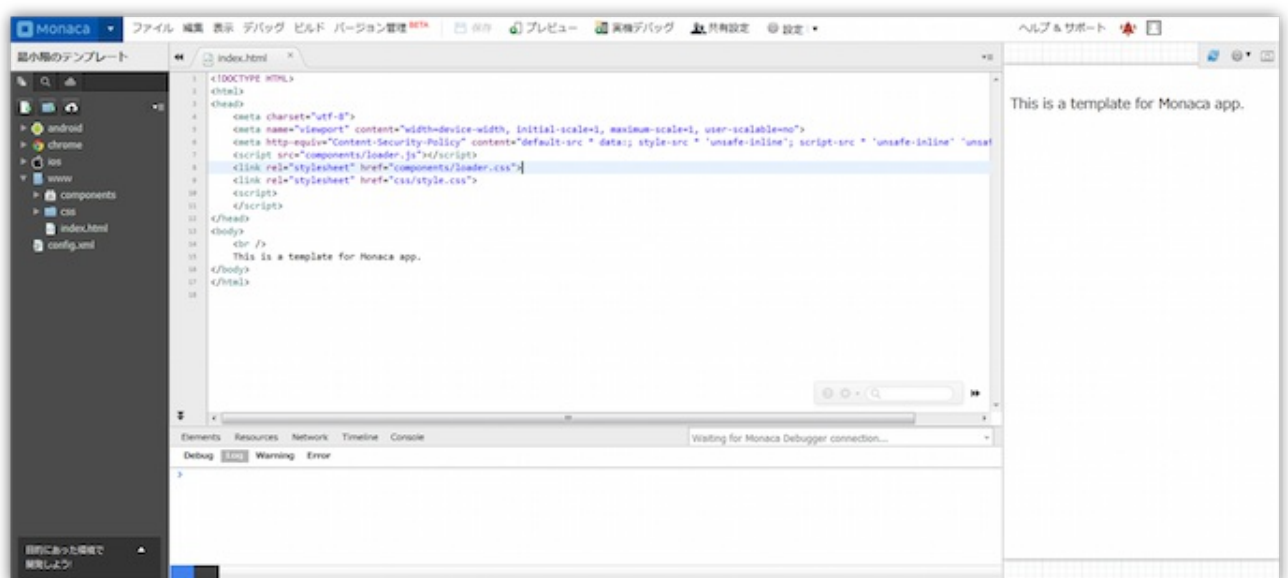


- APIキーは後ほどXcodeアプリで使います

- Monacaにログインし、プロジェクトを作成します
 - はじめての方は、上記リンクより会員登録（無料）を行ってください。
- 「新規プロジェクトの作成」 > 「No Framework」を選択して、最小限のテンプレートを「作成」をクリックします
- 「新規プロジェクト」画面で、プロジェクト名と説明を入力（任意）し、「プロジェクトを作成する」をクリックします



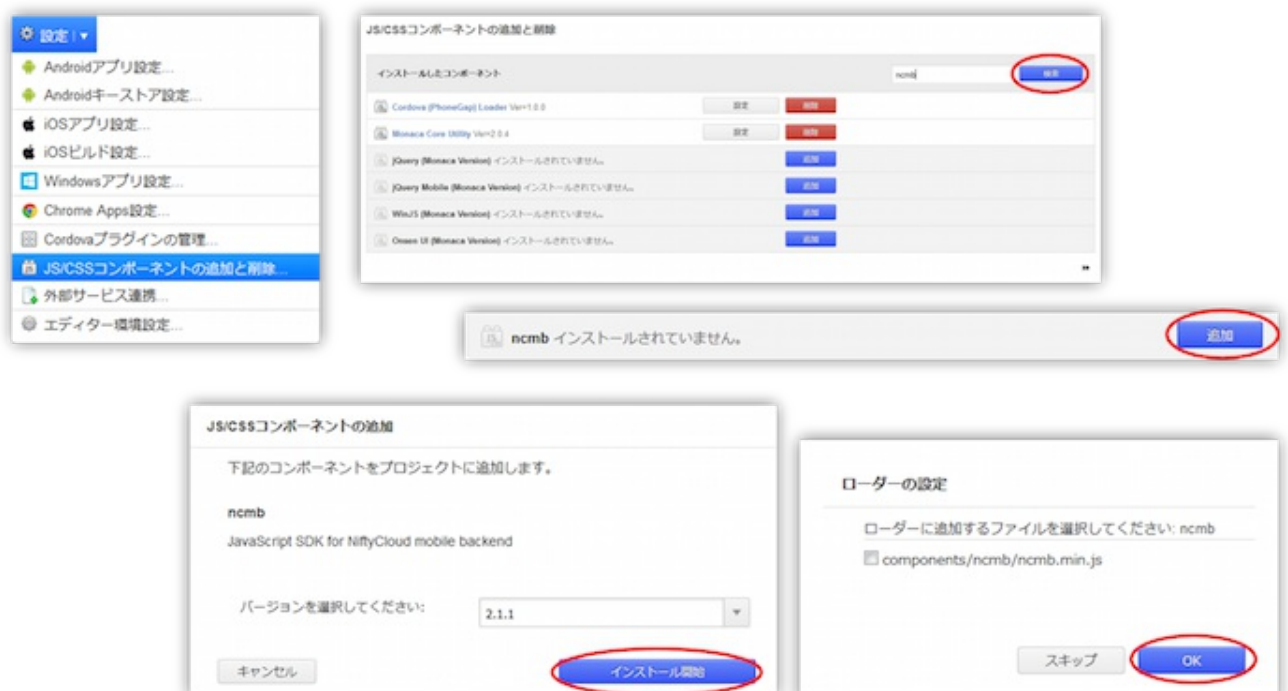
- プロジェクトが作成されたら「開く」をクリックすると、開発環境が表示されます



SDKのインストールと読み込み

Monaca

- 「設定」 > 「JS/CSSコンポーネントの追加と削除...」をクリックします
- 「JS/CSSコンポーネントの追加と削除」画面で「ncmb」と入力して「検索」をクリックします
- 「ncmb インストールされていません。」と出ますので、「追加」をクリックします
- 「JS/CSSコンポーネントの追加」アラート画面が出ますので、「インストール開始」をクリックします
 - バージョンはデフォルトで最新版が選択されていますので、操作は不要です。
 - 画像は最新版のバージョンがv2.1.1の場合です。
- 「ローダーの設定」アラート画面が出ますので、「components/ncmb/ncmb.min.js」にチェックを入れ、「OK」をクリックします



- 「ncmb」が下図のように追加されればSDKのインストールが完了です



APIキーの設定とSDKの初期化

Monaca

- コードを書いていく前に、必ずmBaaSで発行されたAPIキーの設定とSDKの初期化を行う必要があります
- `index.html` の `<script>` と `</script>` の間に次のコードを書きます

```
<script>
// APIキーの設定とSDK初期化
var ncmb = new NCMB("YOUR_APPLICATIONKEY","YOUR_CLIENTKEY");
</script>
```

- 上の「YOUR_APPLICATION_KEY」と「YOUR_CLIENT_KEY」は、mBaaSのダッシュボードで「アプリの新規作成」を行ったときに発行されたAPIキーに置き換えます
 - アプリ作成時のAPIキー発行画面を閉じてしまった場合は、「アプリ設定」>「基本」で確認できます。
 - 「コピー」ボタンを使用してコピーしてください。



- これで連携作業は完了です！サンプルコードを書いて実際にmBaaSを使ってみましょう

サンプルコードの実装

Monaca

- `index.html` の `<script>` と `</script>` の間に書いた処理は、アプリの起動時に実行されます
- APIキーの設定とSDK初期化コードの下にサンプルコードを書くと、すぐに動作確認が可能です

```
<script>
// APIキーの設定とSDK初期化
var ncmb = new NCMB("YOUR_APPLICATIONKEY","YOUR_CLIENTKEY");
// ↓ここにサンプルコードを実装↓

</script>
```

サンプルコード（データストア）

- 次のコードはmBaaSのデータストアに保存先の「TestClass」というクラスを作成し、「message」というフィールドへ「Hello, NCMB!」というメッセージ（文字列）を保存するものです

```
// 保存先クラスの作成
var TestClass = ncmb.DataStore("TestClass");

// 保存先クラスのインスタンスを生成
var testClass = new TestClass();

// 値を設定と保存
testClass.set("message", "Hello, NCMB!")
    .save()
    .then(function(object){
        // 保存に成功した場合の処理
    })
    .catch(function(err){
        // 保存に失敗した場合の処理
    });
```


アプリを実行してmBaaSのダッシュボードを確認する

- 「保存」をクリックして保存します
- アプリを実行します
 - ブラウザ画面上で実行する場合は「プレビュー」をクリックします
 - [Monaca](#)デバッガーで実行する場合は「実機デバッグ」をクリックします

mBaaS ダッシュボード

- アプリが起動されたら、mBaaSのダッシュボードで「データストア」から、データが保存されていることを確認できます



The screenshot shows the mBaaS dashboard interface. The top navigation bar includes links for 'mobile backend', 'アプリ一覧', 'ダッシュボード', 'ドキュメント', '開発TIPS', 'コミュニティ', and '連携サービス'. The main header has buttons for '+新しいアプリ', '(<) 更新', 'test', and 'データストア'. The left sidebar contains menu items: 'メニューを閉じる', 'ダッシュボード', '会員管理', 'データストア', 'ファイルストア', 'スクリプト', and 'プッシュ通知'. The main content area shows the 'TestClass' data store. It includes a table with columns: 'objectId', 'message', 'createDate', and 'updateDate'. A single record is displayed with the following values: 'ycY4U9ZIoZV5eIjQ', 'Hello, NCMB!', '2017-01-10T18:10:18.109+09:00', and '2017-01-10T18:11:18.109+09:00'.

	objectId	message	createDate	updateDate
<input type="checkbox"/>	ycY4U9ZIoZV5eIjQ	Hello, NCMB!	2017-01-10T18:10:18.109+09:00	2017-01-10T18:11:18.109+09:00