Android クイックスタート

2017/05/15作成

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

目次

- アプリの新規作成
- SDKのダウンロード
- SDKのインストール
- AndroidManifest.xml の編集
- SDKの読み込み
- 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

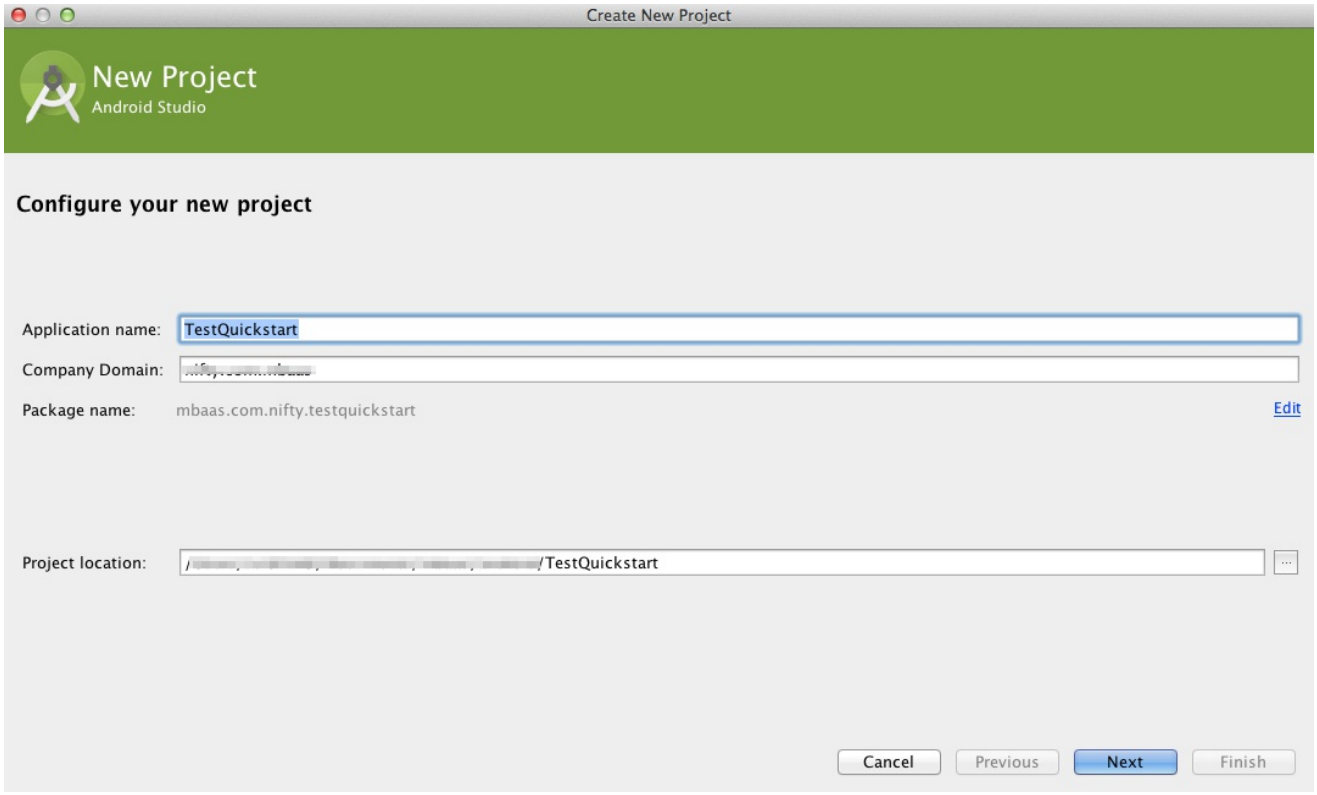
- ニフティクラウドmobile backendにログインします
- ダッシュボードが表示されたら、「アプリの新規作成」を行います
    - すでに別のアプリを作成済みの場合は、ヘッダーの「＋新しいアプリ」をクリックします

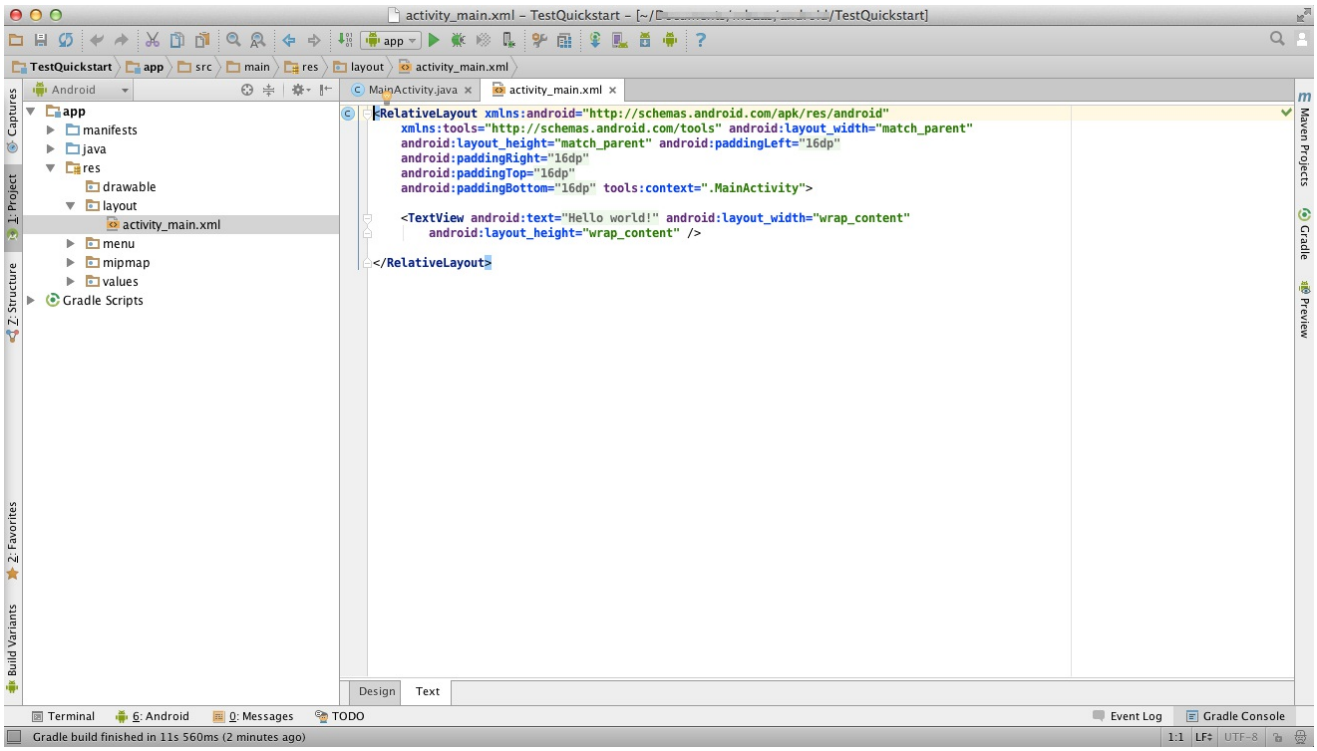
- 「アプリ名」を入力し「新規作成」をクリックすると、APIキー（アプリケーションキーとクライアントキー）が発行されます

- APIキーは後ほどAndroidアプリで使います

Android Studio

- AndroidStudioでプロジェクトを作成します
    - 既存のプロジェクトを利用する場合はこの作業は不要です





SDKのダウンロード

- Githubリリースページの NCMB.x.x.x.zip ボタンからダウンロードしてください
  - 最新版をダウンロードしてください。
  - zipファイルの中身に、NCMB.jarがあります。

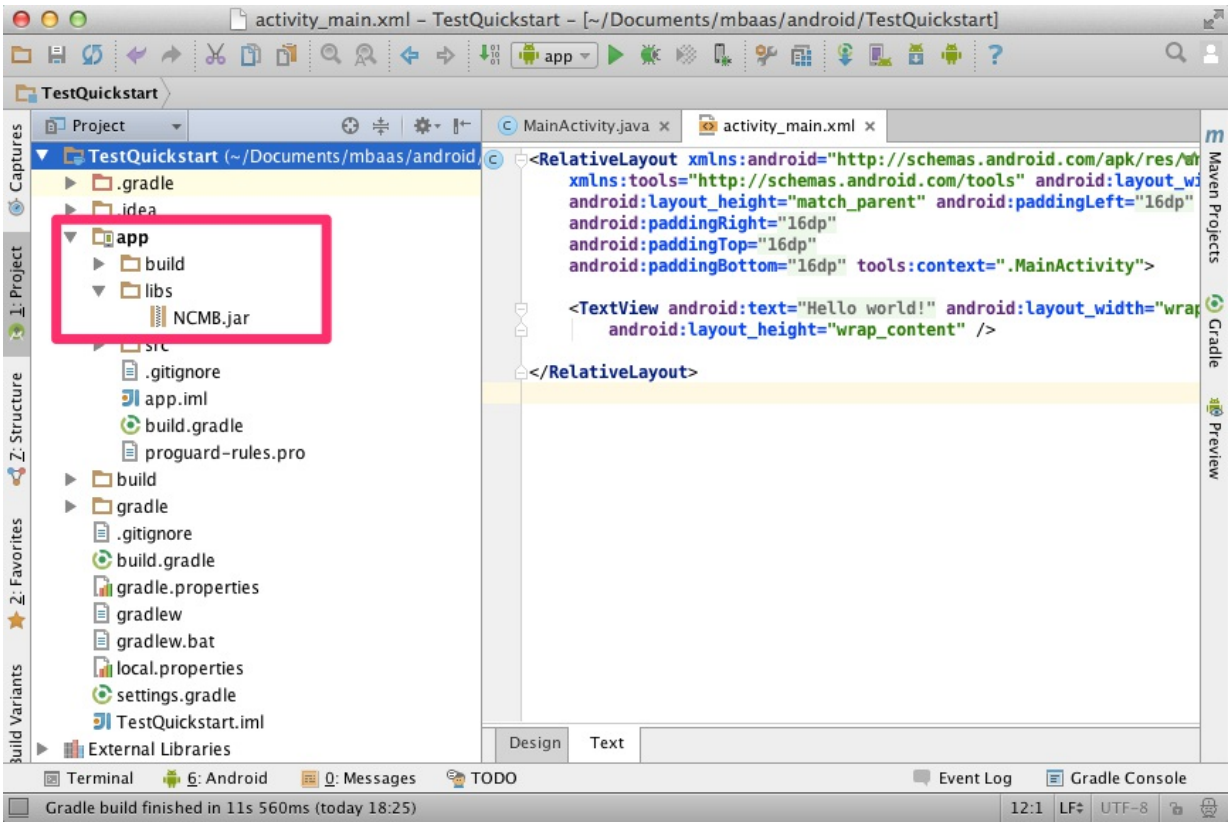
SDKのインストール

- このSDKでは、以下のライブラリを使用しています。
  - **Gson**

Android Studio

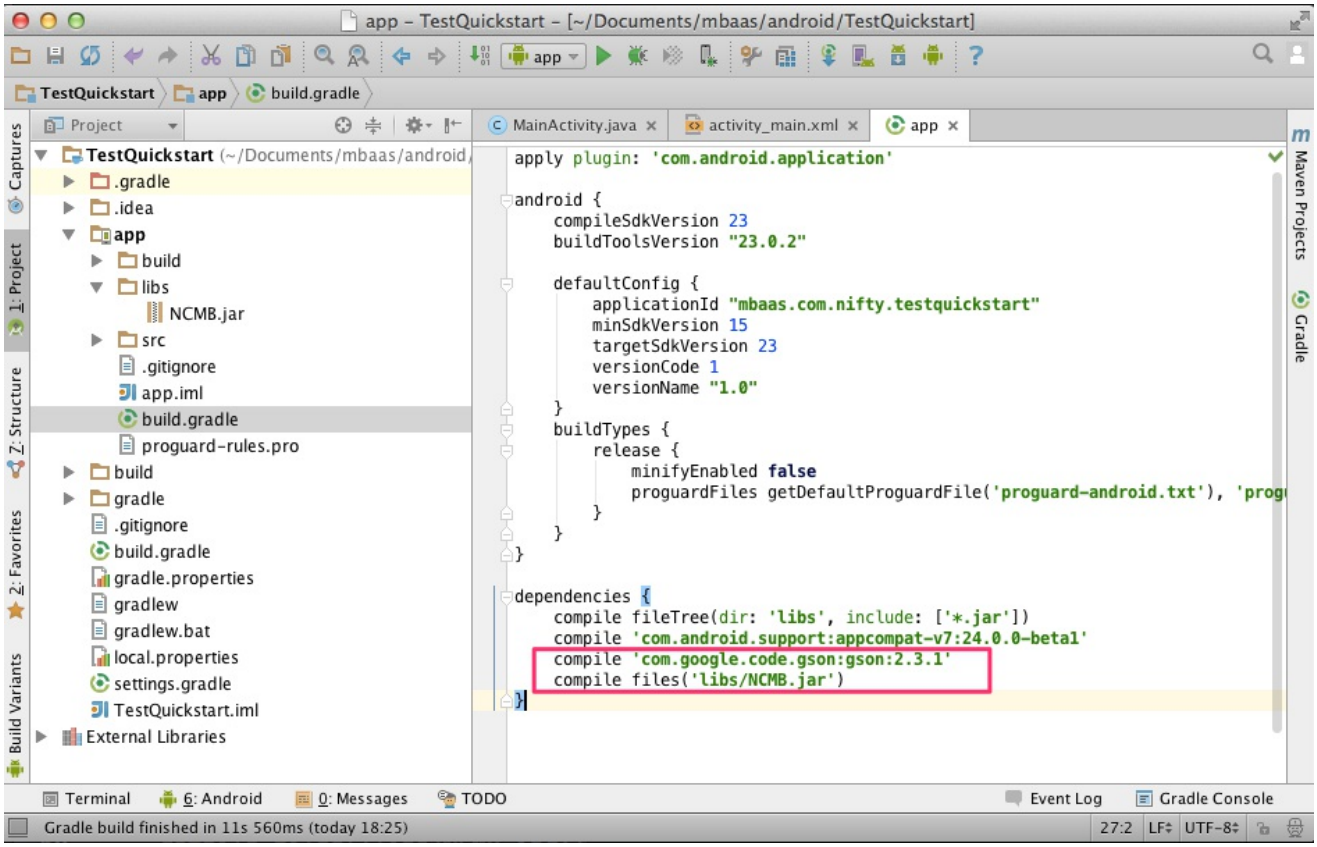
- Android Studioでプロジェクトを開き、以下の手順でSDKをインストールしてください

  - app/libsフォルダにNCMB.jar をコピーします



- app/build.gradleファイルに以下を追加します

```
dependencies {
    compile 'com.google.code.gson:gson:2.3.1'
    compile files('libs/NCMB.jar')
}
```



AndroidManifest.xmlの編集

Android Studio

- <application>タグの直前に以下のpermissionを追加します

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

- プロセスが存在しない状態からサービスなどでアプリを起動しAPIリクエストを行うためには、 `<application>` タグに以下を追加します。

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

SDKの読み込み

Android Studio

- MainActivity.javaの冒頭に次のコードを追記して、インストールしたSDKを読み込みます

```
import com.nifty.cloud.mb.core.NCMB;
```

APIキーの設定とSDKの初期化

Android Studio

- ニフティクラウド mobile backendのアプリケーションキーとクライアントキーを利用して、 NCMBクラスのinitializeメソッドでAndroid SDKの初期化を行います
- アプリ起動時に表示するアクティビティのonCreateメソッドに下記の内容を追記します

```
NCMB.initialize(this.getApplicationContext(),"APP_KEY","CLIENT_KEY");
```

mBaaS
ダッシュボード

- 上の「`APP_KEY`」と「`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

Android Studio

- 最初に、ファイルの先頭に利用するライブラリを追記します

```
import com.nifty.cloud.mb.core.NCMB;
import com.nifty.cloud.mb.core.NCMBException;
import com.nifty.cloud.mb.core.NCMBObject;
```

- `MainActivity.java` の `onCreate` メソッド内に書いた処理は、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

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // APIキーの設定とSDK初期化
    NCMB.initialize(this.getApplicationContext(), "APP_KEY", "CLIENT_KEY");
    // ↓ここにサンプルコードを実装↓


    setContentView(R.layout.activity_main);
}
```

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。

```
// クラスのNCMBObjectを作成
NCMBObject obj = new NCMBObject("TestClass");

// オブジェクトの値を設定
obj.put("message", "Hello, NCMB!");

// データストアへの登録
obj.saveInBackground(new DoneCallback() {
    @Override
    public void done(NCMBException e) {
        if(e != null){
            //保存に失敗した場合の処理

        }else {
            //保存に成功した場合の処理

        }
    }
});
```

アプリを実行してmBaaSのダッシュボードを確認する

- アプリを実機またはシミュレーターで実行します

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