

Android Studioに

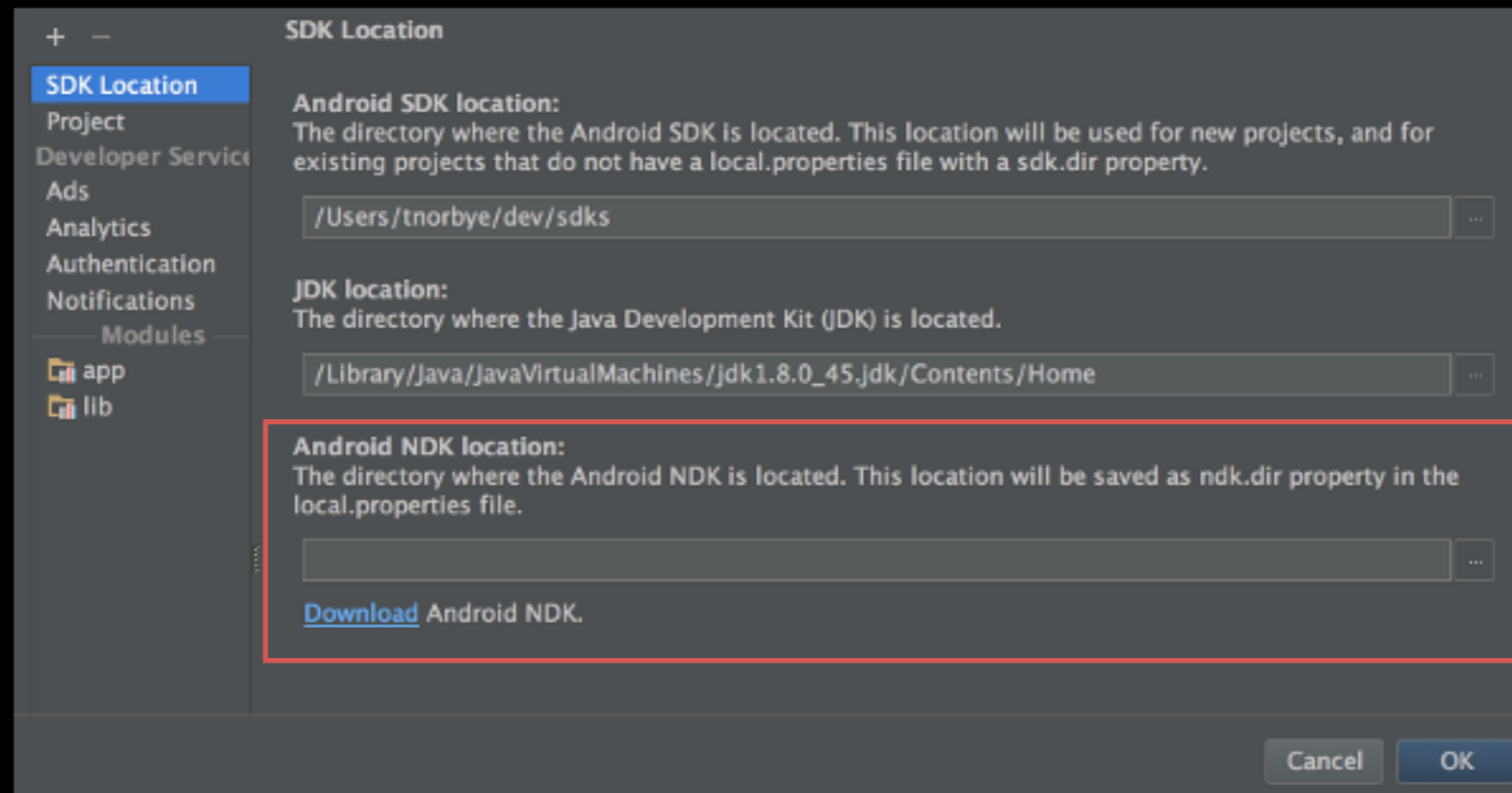
NDK導入

藤田 朋子

Android NDKのダウンロード

「Android Studio>Preferences..」
(ショートカット : command + ;) を開き、
左のタブのSDK Locationを選択する

Android NDKのダウンロード



Android NDKのダウンロード

ファイルがまだダウンロードできていないとき、

[Download](#) Android NDK

を押す

ファイルがダウンロード済みのとき、

Android SDKからAndroid NDKが

最新かどうか確認できる

(ダウンロードできていたらとりあえずOK!)

build.gradle(project)の書き換え

buildscriptのdependenciesを書き換え

build.gradle(project)

```
buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle-experimental:0.7.0-alpha4'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

build.gradle(project)の書き換え

この書き換えでgradleの書式が変わるので、
build.gradle(app)を書き換えないとビルドできないようになり
ます

build.gradle(app)の書き換え

- apply pluginを書き換える
- android{...}をmodel{...}で囲む必要がある
- android{...}の中身の書式を変更する
- 使用するc/c++ファイルを「moduleName “ファイル名”」で指定する

build.gradle(app)

```
apply plugin 'com.android.model.application'

model {
    android {
        compileSdkVersion 23
        buildToolsVersion "23.0.1"

        defaultConfig {
            applicationId "com.test.ndktest"
            minSdkVersion.apiLevel 15
            targetSdkVersion.apiLevel 23
            versionCode 1
            versionName "1.0"

            buildConfigFields {
                create() {
                    type "int"
                    name "VALUE"
                    value "1"
                }
            }
        }
        buildTypes {
            release {
                minifyEnabled false
                proguardFiles.add(file("proguard-rules.pro"))
            }
        }
        ndk {
            moduleName "native"
        }
    }
}
```

忘れないように！

C/C++ファイルを作成

ネイティブファイルの作成方法と
ネイティブコードの呼び出し (Java→C/C++)

c/c++ファイルの作成（方法1）

c/c++ファイルを使うActivityクラスに追加

```
static {  
    System.loadLibrary("native");  
}  
  
public native String createString();
```

createString上にカーソルがある状態で、

「alt + enter」

「Create function …」を選択する

c/c++ファイルの作成（方法1）

「native.c」というファイルが作成されて、
中身は以下のようなになると思います

```
#include <jni.h>

JNIEXPORT jstring JNICALL
Java_com_test_ndktest_MainActivity_createString(JNIEnv *env, jobject instance) {
    return (env)->NewStringUTF("hello JNI");
}
```

この関数名の命名には規則があるので
変えないようにしましょう

c/c++ファイルの作成（方法2）

Android Studioの左のディレクトリ構造を
「Android」にしておく
「app」を右クリックして
「New>Folder>JNI Folder」を選択し、
JNIフォルダを作成

c/c++ファイルの作成（方法2）

先ほど作ったjniフォルダを右クリック

「New>C/C++ Source File」

ファイル名と拡張子を指定してファイル作成

native.cpp (例)

※build.gradle(app)のmoduleNameに
“native”を追加しないと動きません※

```
#include <jni.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

extern "C" {
    JNIEXPORT jstring JNICALL
    Java_com_test_ndktest_MainActivity_createString(JNIEnv *env, jobject instance) {
        return (env)->NewStringUTF("hello JNI");
    }
}
```

C/C++ファイルを作成 (C/C++→Java)

ネイティブコードの呼び出し (C/C++→Java)

C/C++ファイルを作成 (C/C++→Java)

呼び出す関数を定義しているクラスを取得
取得したクラスから関数IDを取得
CallMethodで関数を呼び出す

C/C++ファイルを作成 (C/C++→Java)

Javaファイル

```
public void hello() {  
    //// TODO:  
}
```

C/C++ファイルを作成 (C/C++→Java)

ネイティブファイル (C++)

```
//クラスを見つける  
jclass cls = env->FindClass("com/test/ndktest/MainActivity");  
  
//関数IDを取得  
jmethodID mid = env->GetMethodID(cls, "hello", "()V");  
  
//関数呼び出し  
env->CallVoidMethod(instance, mid);
```

文字色を黒にするのが面倒くさくなったので・・・

ごめんなさい

C/C++ファイルを作成 (C/C++→Java)

```
jclass JNIEnv::FindClass(const char *name);
```

引数

name : 完全修飾クラス名または配列型シグニチャー

戻り値

クラスオブジェクト。失敗時はNULL

C/C++ファイルを作成 (C/C++→Java)

```
jmethodID JNIEnv::GetMethodID(jclass clazz,  
    const char *name, const char *sig);
```

引数

clazz : クラスオブジェクト

name : メソッド名

sig : メソッドシグニチャー

戻り値

メソッドID。失敗時はNULL

C/C++ファイルを作成 (C/C++→Java)

取得したクラスから関数IDを取得するとき、
Javaのメソッドシグネチャ（シグニチャー）が必要！

シグネチャを確認するには、

コンソールを開いて、

「project\$ javap -s <クラスファイルのパス>」

(私の環境でのパスは「app/build/intermediates/
classes/debug/com/test/ndktest/
MainActivity.class」でした。)

C/C++ファイルを作成 (C/C++→Java)

```
void JNIEnv::CallVoidMethod(jobject obj,  
                             jmethodID methodID, ...);
```

引数

obj : JNI関数のjobject引数を指定

methodID : 呼び出す関数のメソッドID

... : 呼び出す関数の引数

C/C++ファイルを作成 (C/C++→Java)

※注意※

ネイティブスレッドからJavaの関数を呼び出すときは、
JavaVMを取得しなければいけません。
(そういう設計にするかわからないので、
説明は端折ります。
必要であれば、聞いてください)

おわり

ざっくりとした説明なので、
詰まるところもあるかと思えます。
Slackやtwitter (@cndtmc) のメンションで
質問していただければお答えします！

参考にしたサイト

Android NDK Preview

[http://tools.android.com/tech-docs/android-ndk-
preview](http://tools.android.com/tech-docs/android-ndk-preview)

参考にしたサイト

AndroidでネイティブからJavaを呼び出す

<http://hirooom2.jimdo.com/2015/06/11/android>

[%E3%81%A7%E3%83%8D](http://hirooom2.jimdo.com/2015/06/11/android)

[%E3%82%A4%E3%83%86%E3%82%A3%E3%83](http://hirooom2.jimdo.com/2015/06/11/android)

[%96%E3%81%8B%E3%82%89java](http://hirooom2.jimdo.com/2015/06/11/android)

[%E3%82%92%E5%91%BC](http://hirooom2.jimdo.com/2015/06/11/android)

[%E3%81%B3%E5%87%BA%E3%81%99/](http://hirooom2.jimdo.com/2015/06/11/android)