链滴

Android 现有工程添加 C/C++ 支持

作者：RustFisher

原文链接：https://ld246.com/article/1599121113505

来源网站：链滴

添加C/C++支持

原文链接

前面我们新建工程时选择支持C/C++。如何给现有项目添加C/C++支持呢?

添加步骤

以Tutorial2020工程为例。我们给它手动添加NDK。
工程中有一个**app**模块。往这个模块中添加NDK。

添加cpp目录

在**app/src/main**目录下，新建**cpp**目录，其与**java**同级。

在**cpp**目录中新建文件fisher-lib.cpp。目前这个文件还是空的，后面再添加代码。

新建native方法

新建CalUtil类，里面定义native方法。

```
package com.rustfisher.tutorial2020.cal;

public class CalUtil {
    public native int getNumber();

    public native String getMsg();
}
```

添加cmake配置

在**cpp**目录中新建CMakeLists.txt文件。
使用add_library方法进行配置。库名叫做fisher-lib。

```
cmake_minimum_required(VERSION 3.4.1)
add_library( # Specifies the name of the library.
             fisher-lib

             # Sets the library as a shared library.
             SHARED

             # Provides a relative path to your source file(s).
             fisher-lib.cpp )
```

此时cmake还没有起作用。从as中可以看出字体都还是灰色。我们还需要配置gradle。

配置gradle

仿照之前的新建工程，添加gradle配置externalNativeBuild。指定cmake的路径和版本。

```
android {

    defaultConfig {
        // 要添加的部分
        externalNativeBuild {
            cmake {
                cppFlags ""
            }
        }
    }

    // 要添加的部分
    externalNativeBuild {
        cmake {
            path "src/main/cpp/CMakeLists.txt"
            version "3.10.2"
        }
    }
}
```

sync后可以发现CMakeLists.txt的代码已经有颜色了。

实现cpp

现在gradle和cmake已经配置完毕，我们来实现cpp代码。

CalUtil中有2个native方法，方法签名前缀是Java_com_rustfisher_tutorial2020_cal_CalUtil_。

当我们输入g时，as会智能提示出我们想要的方法名。十分便利。

```
#include <jni.h>
#include <string>

extern "C" JNIEXPORT jstring JNICALL
Java_com_rustfisher_tutorial2020_cal_CalUtil_getMsg(JNIEnv *env, jobject thiz) {
    std::string hello = "欢迎来到Tutorial";
    return env->NewStringUTF(hello.c_str());
}

extern "C" JNIEXPORT jint JNICALL
Java_com_rustfisher_tutorial2020_cal_CalUtil_getNumber(JNIEnv *env, jobject thiz) {
    return 2020;
}
```

这里遇到一个as的问题，编写好cpp代码后，会有红线提示代码错误。但是直接运行项目又是正常的。

重新打开一次这个工程，红线就不见了。点击左边的图标，可以跳转到对应的Java方法。

```cpp
//
// Created by RustFisher on 2020/5/28.
//
#include <jni.h>
#include <string>

extern "C" JNIEXPORT jstring JNICALL
Java_com_rustfisher_tutorial2020_cal_CalUtil_getMsg(JNIEnv *env, jobject CalUtil thiz) {
    std::string hello = "欢迎来到Tutorial";
    return env->NewStringUTF(hello.c_str());
}

extern "C" JNIEXPORT jint JNICALL
CalUtil.getNumber(JNIEnv *env, jobject CalUtil thiz) {
    return 2020;
}
```

调用方法

调用这2个方法。

```
private CalUtil mCalUtil = new CalUtil();
    // ....
    mCalUtil.getMsg();
    mCalUtil.getNumber()
```

至此给现有项目添加NDK完毕。

小结

- 添加 **cpp**目录
- 在 **cpp**目录中添加cmake
- 在gradle中配置cmake
- 新建native方法
- 编写 **cpp**实现方法
-

原文链接