

如果按照上述方法显示找不到定义，可能是以下原因之一导致的。以下是排查步骤和解决方案：

1. 检查 JNI 方法签名是否正确

JNI 方法的签名必须严格按照规则命名：

Java_<全限定类名>_<方法名>

例如：

```
package com.example.audio;

public class AudioNative {
    public native void
    initAudioEngine();
}
```

对应的 JNI 方法名应为：

```
extern "C"
JNIEXPORT void JNICALL
Java_com_example_audio_AudioNative_initAudioEngine(JNIEnv *env, jobject
thiz) {
    // 方法实现
}
```

解决方法：

- 确认 `com.example.audio.AudioNative` 的包名和类名与 JNI 方法名一致。
- 如果包名不一致，修改 `Java_com_example_audio_AudioNative_initAudioEngine` 为正确的类路径。

2. 确保 `System.loadLibrary` 加载了正确的库

Java/Kotlin 代码中需要加载与 CMake 定义的 `native-lib` 名称一致的库：

```
static {  
    System.loadLibrary("native-lib");  
}
```

检查：

- 确认 `CMakeLists.txt` 中的库名：

```
add_library(  
    native-lib  
    SHARED  
    native-lib.cpp  
)
```

- 如果库名不匹配，修改 `System.loadLibrary("native-lib");` 中的名字为实际的库名。

3. 检查 CMake 配置和头文件路径

确保 miniaudio.h 包含路径正确

在 CMakeLists.txt 中，添加头文件路径：

```
include_directories($  
{CMAKE_SOURCE_DIR}/native/miniaudio)
```

确认：

- miniaudio.h 文件是否放在 native/miniaudio 目录下。
- 如果路径不对，调整 include_directories。

确保 C++ 文件被编译

确认 CMakeLists.txt 是否正确包含了源文件，例如：

```
add_library(  
    native-lib  
    SHARED  
    native-lib.cpp # 确保这个文件被列出
```

)

如果有多个文件，确保所有文件都被列出。

4. 检查是否重建了项目

步骤：

- 在 Android Studio 中点击 File > Sync Project with Gradle Files。
- 确保 CMakeLists.txt 的修改被同步。
- 重新构建项目 (Build > Rebuild Project)。

5. 确认库正确打包

确认 native-lib 被打包到最终的 APK 中：

1. 在 Android Studio 中打开 build 目录。
2. 检查 lib/<架构> 目录（如 lib/arm64-v8a），是否存在 libnative-lib.so。

如果没有：

- 检查 externalNativeBuild 配置是否正确：

```
android {  
    ...  
    externalNativeBuild {  
        cmake {
```

```
        path "CMakeLists.txt"
    }
}
}
```

6. 日志调试

添加日志，检查问题来源：

1. 确保 miniaudio.h 正常被加载：

```
#include "miniaudio.h"
#include <android/log.h>
#define LOG_TAG "AudioNative"
#define LOGI(...)
__android_log_print(ANDROID_LOG_INFO,
LOG_TAG, __VA_ARGS__)

LOGI("miniaudio header loaded
successfully.");
```

2. 确保 JNI 方法被调用：

```
LOGI("initAudioEngine called.");
```

7. 其他可能原因

方法签名大小写错误

- Java 方法名与 C++ 方法名区分大小写。

- 检查 `initAudioEngine` 拼写是否一致。

ABI 不匹配

- 确保 `build.gradle` 中配置了 NDK 支持的架构，例如：

```
ndk {  
    abiFilters "armeabi-v7a", "arm64-v8a", "x86", "x86_64"  
}
```

NDK 工具链版本问题

- 尝试升级或指定 NDK 版本：

```
android {  
    ndkVersion '25.2.9519653' // 示例  
版本  
}
```

总结

通过以下方式定位问题：

1. 确认方法签名。
2. 确保库加载正确。
3. 检查 CMake 配置和头文件路径。
4. 打开日志，检查运行时信息。

如仍有问题，请提供具体错误日志或代码结构，我会进一步帮助你！