

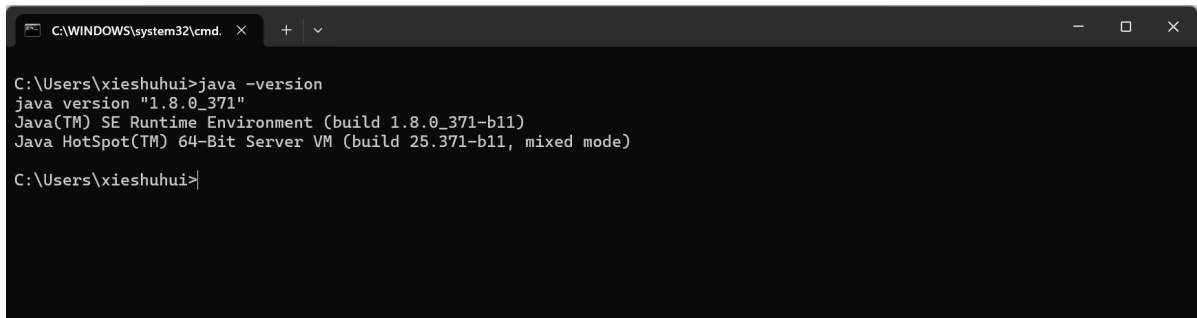
# 安卓app操作文档

## 一、实现目标

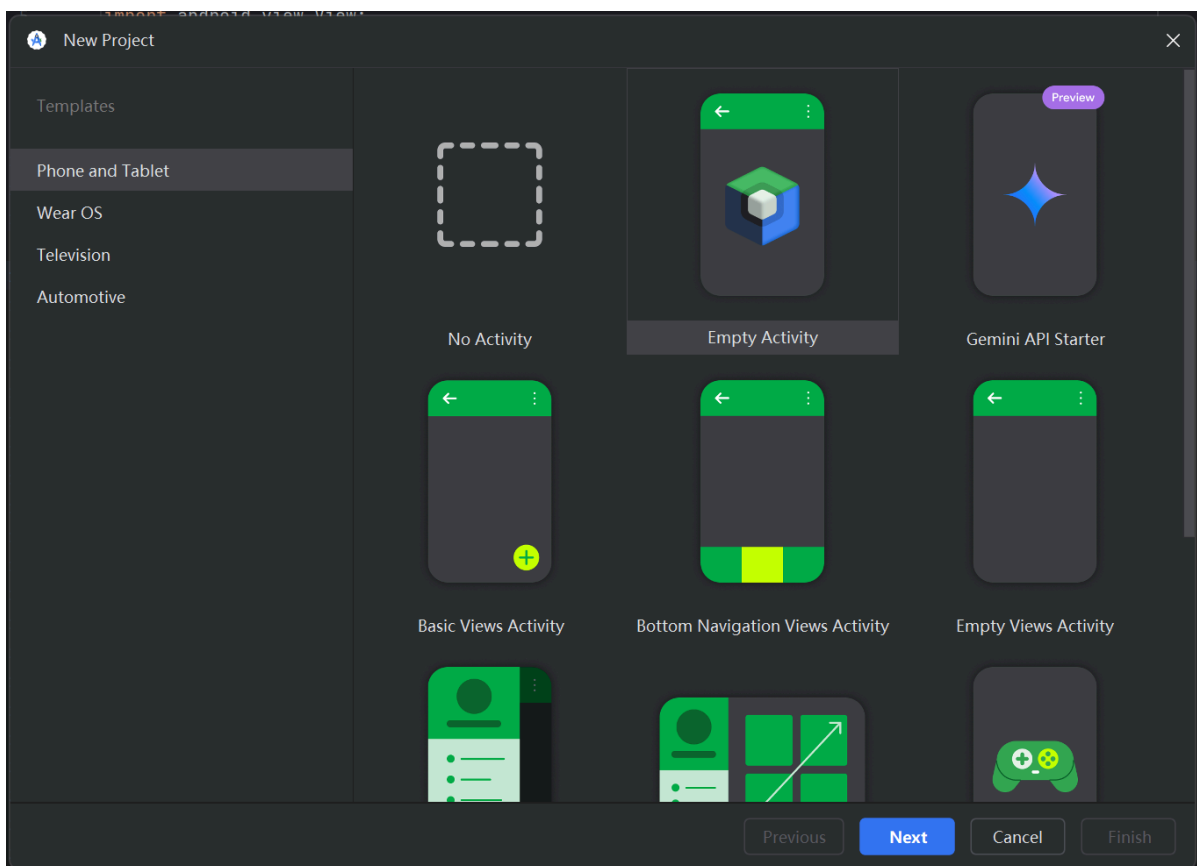
1. 实现页面按钮跳转页面
2. 点击按钮后，页面显示来自JNI的字符串

## 二、环境搭建

### 2.1 jdk安装



### 2.2 andriod studio 安装



## 三、具体实现

### 3.1 设置页面跳转

为了实现跳转页面，首先需要创建两个活动：（MainActivity 主页面）和 SecondActivity（跳转后的页面）

#### 3.11 MainActivity

```
package com.example.myapplication;
```

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btnNavigate = findViewById(R.id.btnNavigate);

        // 设置按钮点击事件, 跳转到SecondActivity
        btnNavigate.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
                SecondActivity.class);
                startActivity(intent);
            }
        });
    }
}

```

### 3.12 SecondActivity

```

package com.example.myapplication;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class SecondActivity extends AppCompatActivity {

    static {
        System.loadLibrary("native-lib"); // 加载JNI库
    }

    // 声明JNI方法
    private native String getJNIString();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        Button btnShowJNI = findViewById(R.id.btnShowJNI);
        TextView textViewFromJNI = findViewById(R.id.jniTextView);
    }
}

```

```

// 设置按钮点击事件，获取并显示JNI字符串
btnShowJNI.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 调用JNI方法并显示结果
        textViewFromJNI.setText(getJNIString());
    }
});
}
}
}

```

## 3.2 集成JNI

### 3.2.1 在 native-lib.cpp 中，编写JNI方法以返回字符串

```

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

extern "C" JNIEXPORT jstring JNICALL
Java_com_example_myapplication_SecondActivity_getJNIString(JNIEnv* env, jobject
/* this */) {
    std::string hello = "Hello from JNI!";
    return env->NewStringUTF(hello.c_str());
}

```

### 3.2.2 CMakeLists.txt编写

```

cmake_minimum_required(VERSION 3.10.2)

# Set project name
project("myapplication")

# Set the path for the source file (make sure the path is correct)
add_library(native-lib SHARED
E:/CodeProject/AndroidProjects/MyApplication/app/src/main/cpp/native-lib.cpp)

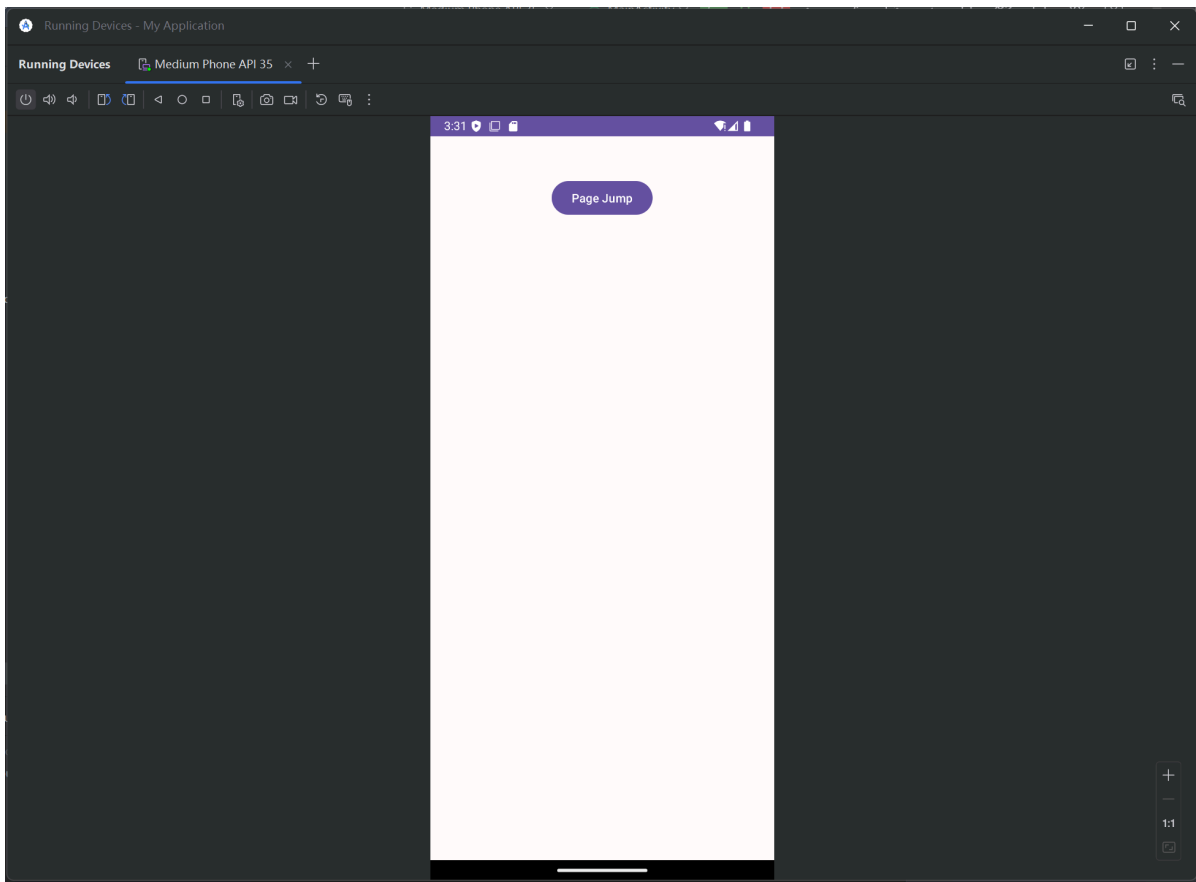
# Find the log library (important for Android projects)
find_library( log-lib
log )

# Link the native library with the log library
target_link_libraries( native-lib
${log-lib} )

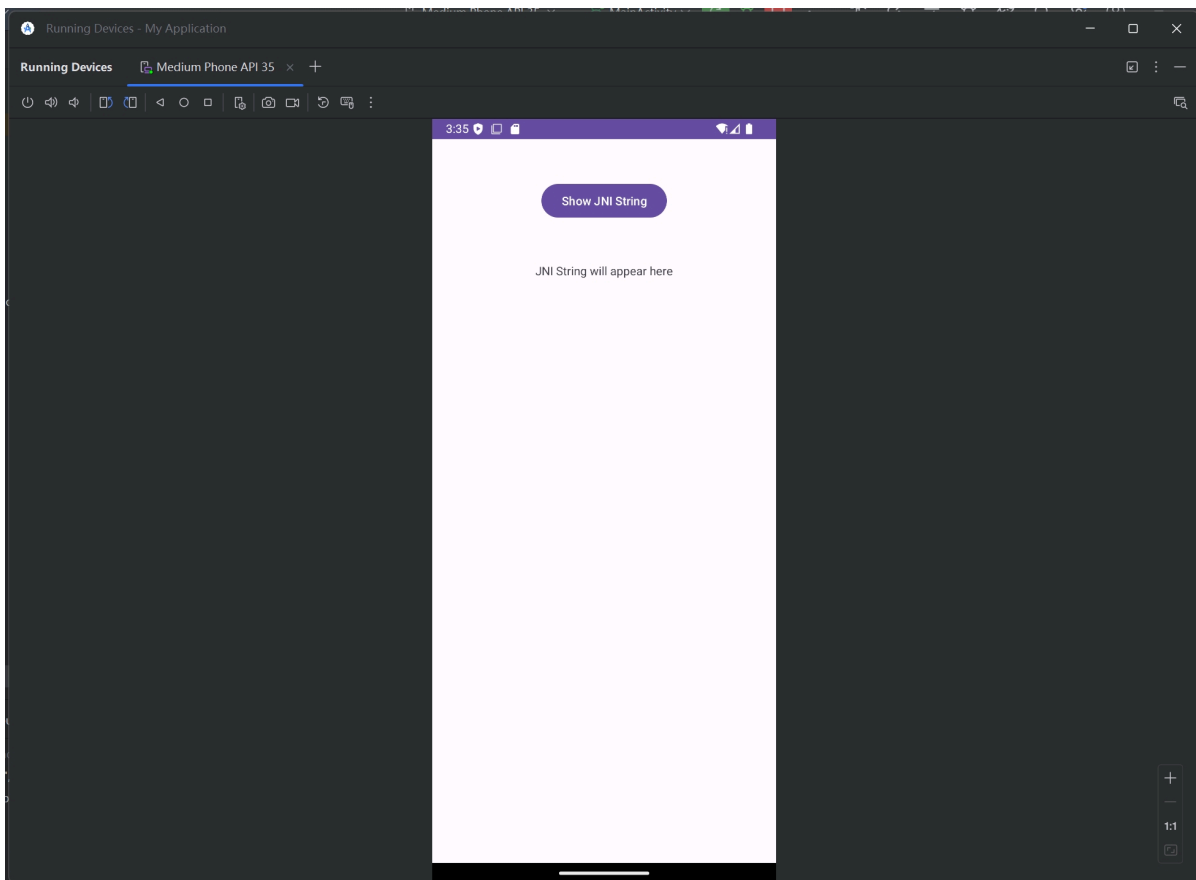
```

## 四、运行结果

### 4.1 初始运行界面



点击Page Jump跳转到第二页面



#### 4.2 验证跳转功能

点击Show JNI String 按钮，成功显示“Hello from JNI!”

