

实验一

班级点名程序实验指导手册

版本：V 1.0

目录

(一) 实验目的	3
(二) 实验涉及知识点	3
(三) 实验准备	3
(四) 实验内容及课时分配	3
(五) 实验过程	4
1. 创建项目	4
2. 修改标题	7
3. 添加显示姓名文本	8
4. 添加点名按钮	9
5. 响应按钮点击事件	11
6. 完成点名逻辑	13
7. 在远程模拟器上运行项目	14

(一) 实验目的

1. 掌握 devEco Studio 的用法
2. 创建项目培养编程思维

(二) 实验涉及知识点

1. HarmonyOS 移动应用开发工具（DevEco Studio）使用
2. UI 组件使用，包括 Text、Button 等
3. TypeScript 基本语法
4. ArkTS 基础语法
5. 按钮监听
6. 代码编程规范（重难点）

(三) 实验准备

1. 技能要求：

操作此实验需要具备基本的 TS 语法知识；

2. 实验环境要求：

基于 Windows10 或者 MacOS 操作系统，安装了 DevEco Studio 开发工具。

(四) 实验内容及课时分配

序号	实验内容	实验课时	对应核心知识点	对应实验目标点
1	页面布局	0.5	1、2、3、4	1、2
2	按钮点击和点名逻辑实现	0.5	3、4、5、6	2
	总计	1		

(五) 实验过程

实验完成功能：

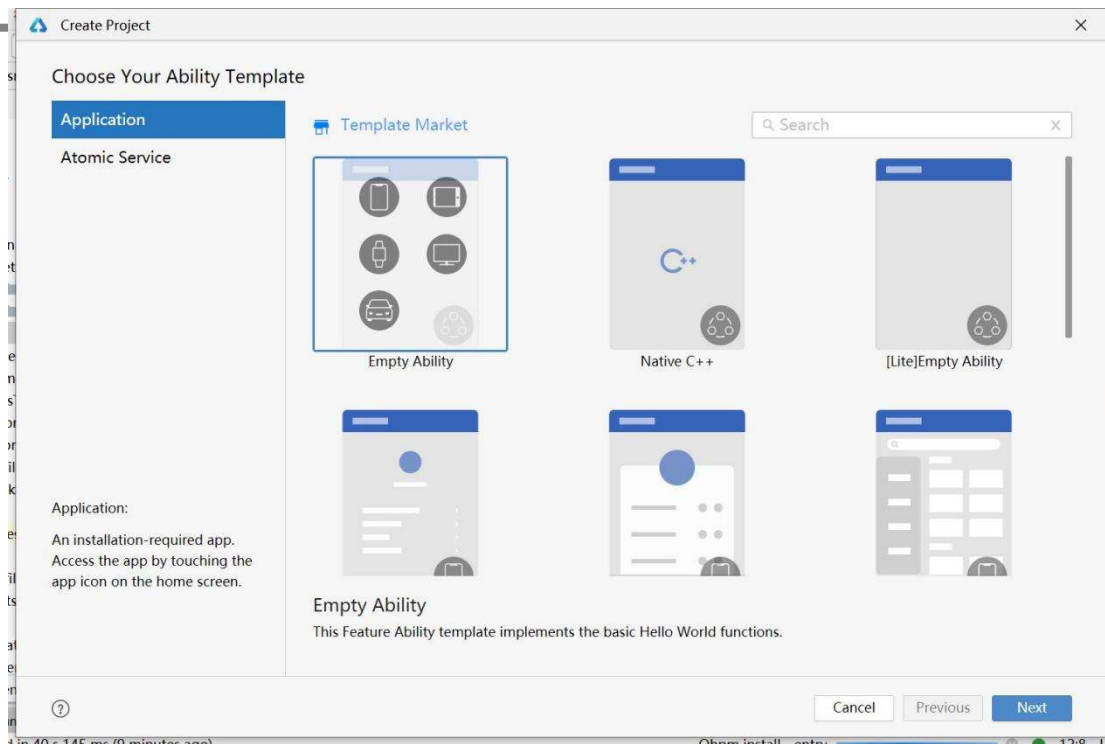
点击提问按钮会随机从班级的点名册中抽取一名同学的名字显示在手机屏幕上。

界面：

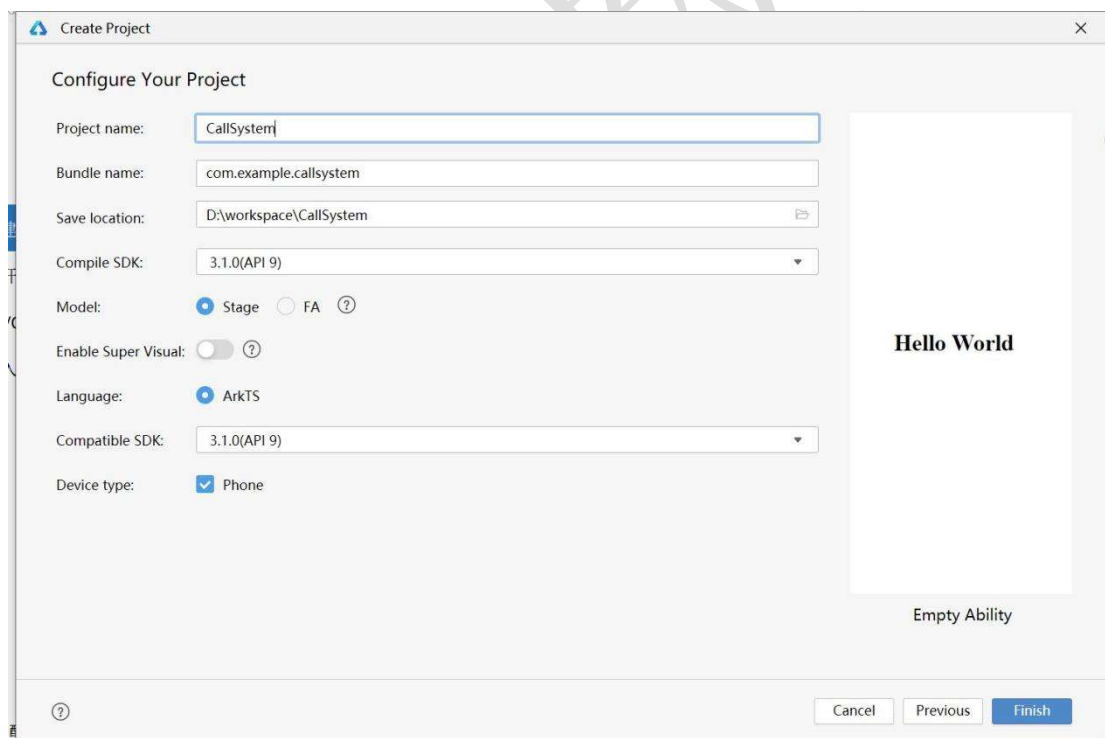


1. 创建项目

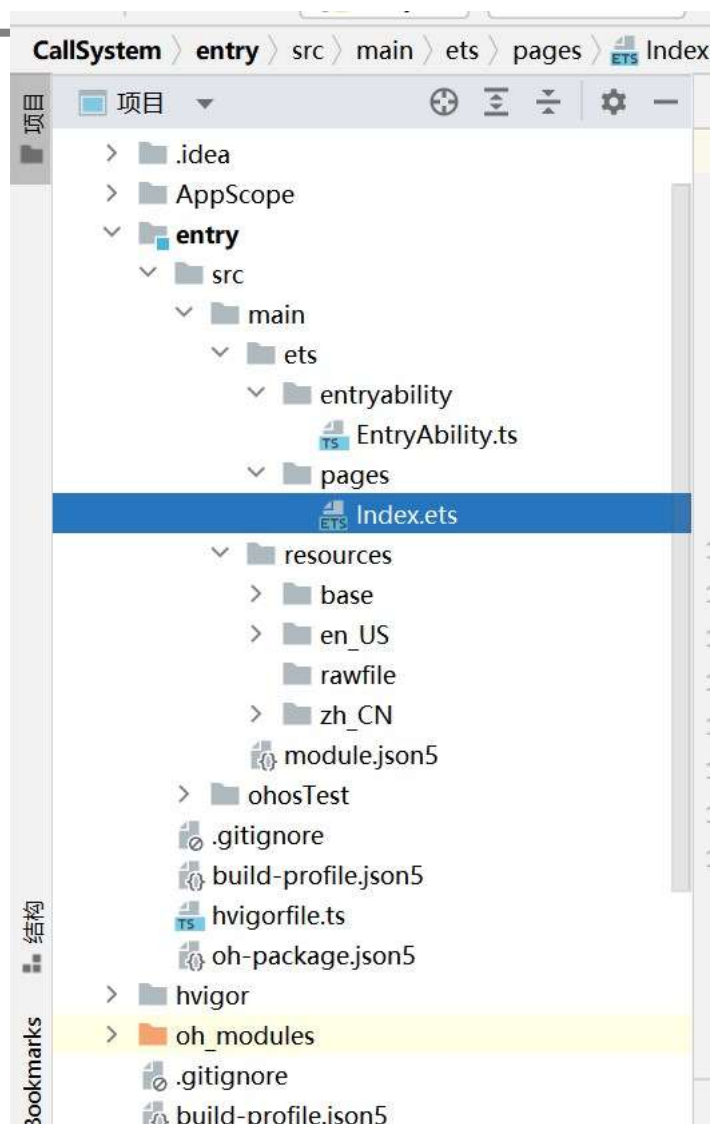
- 打开 DevEco，新建一个 HarmonyOS 项目：



- 配置项目信息：



- 等后台任务运行完成，其项目结构会和下面类似：



项目结构说明：

entry: HarmonyOS 工程模块，编译构建生成一个 HAP 包。

- **src > main > ets:** 用于存放 ets 源码。
- **src > main > ets > entryability:** 应用/服务的入口。
- **src > main > ets > pages:** 应用/服务包含的页面。
- **src > main > resources:** 用于存放应用/服务所用到的资源文件，如图形、多媒体、字符串、布局文件等。关于资源文件，详见资源分类与访问。
- **src > main > module.json5:** 模块配置文件。主要包含 HAP 包的配置信息、应用/服务在具体设备上的配置信息以及应用/服务的全局配置信息。

- **build-profile.json5**: 当前的模块信息、编译信息配置项, 包括 **buildOption**、**targets** 配置等。
 - **hvmigorfile.ts**: 模块级编译构建任务脚本, 开发者可以自定义相关任务和代码实现。
- **build-profile.json5**: 应用级配置信息, 包括签名、产品配置等。
- **hvmigorfile.ts**: 应用级编译构建任务脚本。

2. 修改标题

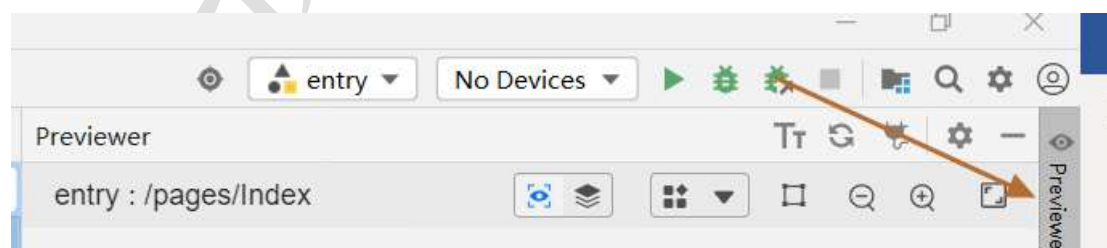
修改 **Index.ets** 文件(目录: **entry > src > main > ets > pages**, 打开“**Index.ets**”文件):

Index.ets 中的代码如下所示:

```
@Entry
@Component
struct Index {
  @State message: string = '班级点名系统'

  build() {
    Row() {
      Column() {
        Text(this.message)
          .fontSize(50)
          .fontWeight(FontWeight.Bold)
      }
      .width('100%')
    }
    .height('100%')
  }
}
```

当前停留在 **Index.ets** 下, 打开预览器:



预览效果:

班级点名系统

3. 添加显示姓名文本

继续添加代码：

```
@State name: string = '姓名：待抽取'  
...  
Text(this.name)  
  .fontSize(50)  
  .fontWeight(FontWeight.Normal)
```

当前 Index.ets 中的完整代码：

```
@Entry  
@Component  
struct Index {  
  @State message: string = '班级点名系统'  
  @State name: string = '姓名：待抽取'  
  build() {  
    Row() {  
      Column() {  
        Text(this.message)  
          .fontSize(50)  
          .fontWeight(FontWeight.Bold)
```



```

        Text(this.name)
            .fontSize(50)
            .fontWeight(FontWeight.Normal)
        }
        .width('100%')
    }
    .height('100%')
}
}

```

在预览器中预览效果：

班级点名系统
姓名：待抽取

4. 添加点名按钮

添加点名按钮：

```

Button() {
    Text('点名')
        .fontSize(50)
        .fontWeight(FontWeight.Normal)
}
.type(ButtonType.Capsule)

```

```

        .margin({
            top: 20
        })
        .backgroundColor('#0D9FFB')
        .width('40%')
        .height('10%')
    }
}

```

当前 Index.ets 中的完整代码：

```

@Entry
@Component
struct Index {
    @State message: string = '班级点名系统'

    build() {
        Row() {
            Column() {
                Text(this.message)
                    .fontSize(50)
                    .fontWeight(FontWeight.Bold)
                Text('姓名：待抽取')
                    .fontSize(50)
                    .fontWeight(FontWeight.Normal)
                Button() {
                    Text('点名')
                        .fontSize(50)
                        .fontWeight(FontWeight.Normal)
                }
                    .type(ButtonType.Capsule)
                    .margin({
                        top: 20
                    })
                    .backgroundColor('#0D9FFB')
                    .width('40%')
                    .height('10%')
            }
            .width('100%')
        }
        .height('100%')
    }
}

```

预览效果：

班级点名系统

姓名：待抽取

点名

5. 响应按钮点击事件

在按钮上绑定点击事件：

```
.onClick((event: ClickEvent) =>{  
    this.name = '刘备'  
})
```

当前 Index.ets 中的完整代码：

```
@Entry  
@Component  
struct Index {  
    @State message: string = '班级点名系统'  
    @State name: string = '姓名：待抽取'  
  
    build() {  
        Row() {  
            Column() {  
                Text(this.message)  
                    .fontSize(50)  
                    .fontWeight(FontWeight.Bold)  
                Text(this.name)  
                    .fontSize(50)
```

```

        .fontWeight(FontWeight.Normal)
    Button() {
        Text('点名')
        .fontSize(50)
        .fontWeight(FontWeight.Normal)
    }
    .type(ButtonType.Capsule)
    .margin({
        top: 20
    })
    .backgroundColor('#0D9FFB')
    .width('40%')
    .height('10%')
    .onClick((event: ClickEvent) =>{
        this.name = '刘备'
    })
}
.width('100%')
}
.height('100%')
}
}

```

点击按钮后的运行效果：



6. 完成点名逻辑

点名逻辑：

- 1) 初始化学生数组
- 2) 根据学生数组长度随机产生下标
- 3) 根据随机下标得到产生的学生姓名
- 4) 设置学生姓名

实现代码：

```
.onClick((event: ClickEvent) =>{  
  
    let names:any = ['刘备','关羽','张飞']  
    let index = Math.floor(Math.random() * 3)  
  
    this.name = names[index]  
})
```

当前 Index.ets 中的完整代码：

```
@Entry  
@Component  
struct Index {  
    @State message: string = '班级点名系统'  
    @State name: string = '姓名：待抽取'  
  
    build() {  
        Row() {  
            Column() {  
                Text(this.message)  
                    .fontSize(50)  
                    .fontWeight(FontWeight.Bold)  
                Text(this.name)  
                    .fontSize(50)  
                    .fontWeight(FontWeight.Normal)  
                Button() {  
                    Text('点名')  
                        .fontSize(50)  
                        .fontWeight(FontWeight.Normal)  
                }  
                .type(ButtonType.Capsule)  
                .margin({  
                    top: 20  
                })  
                .backgroundColor('#0D9FFB')  
            }  
        }  
    }  
}
```

```

        .width('40%')
        .height('10%')
        .onClick((event: ClickEvent) =>{

            let names:any = ['刘备','关羽','张飞']
            let index = Math.floor(Math.random() * 3)

            this.name = names[index]
        })
    }
    .width('100%')
    }
    .height('100%')
    }
}

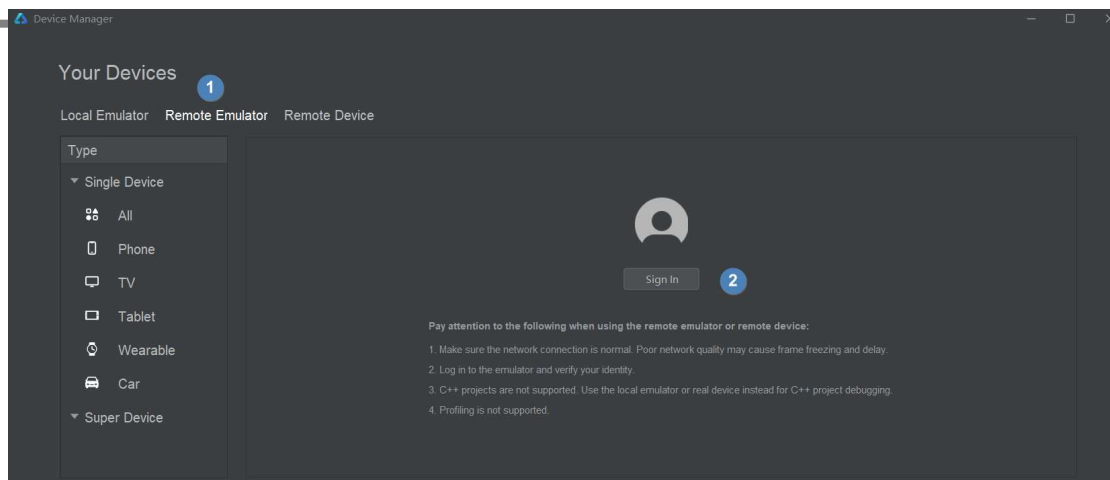
```

点击【点名】按钮后的运行效果：



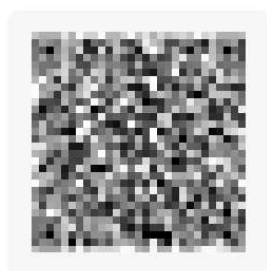
7. 在远程模拟器上运行项目

菜单：tools》Device Manager



需登录华为账号：

华为帐号登录



用华为手机，请前往“设置” > “华为帐号”扫码登录。

[短信验证码登录](#)

[注册](#) | [忘记密码](#) | [遇到问题](#)

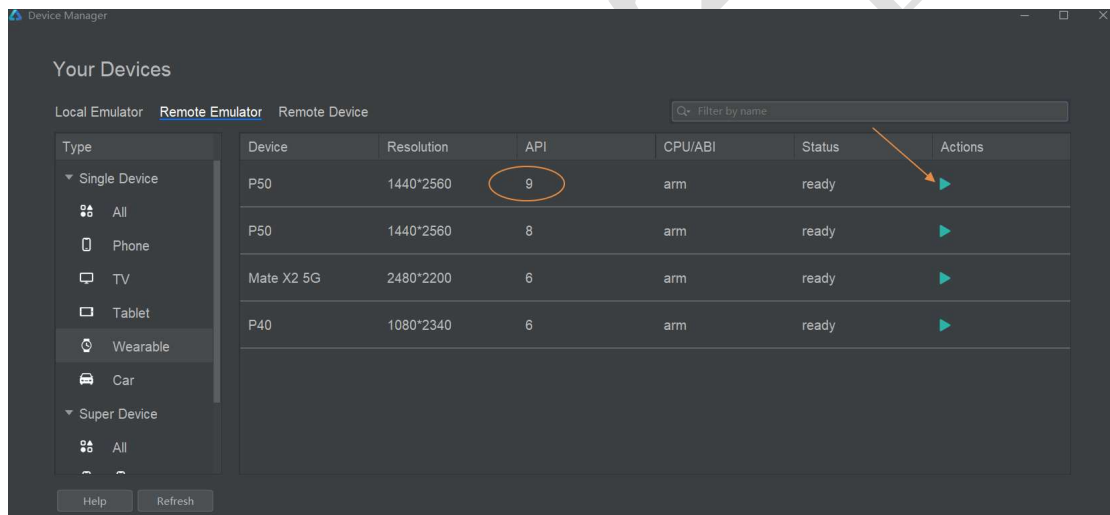
授权成功：



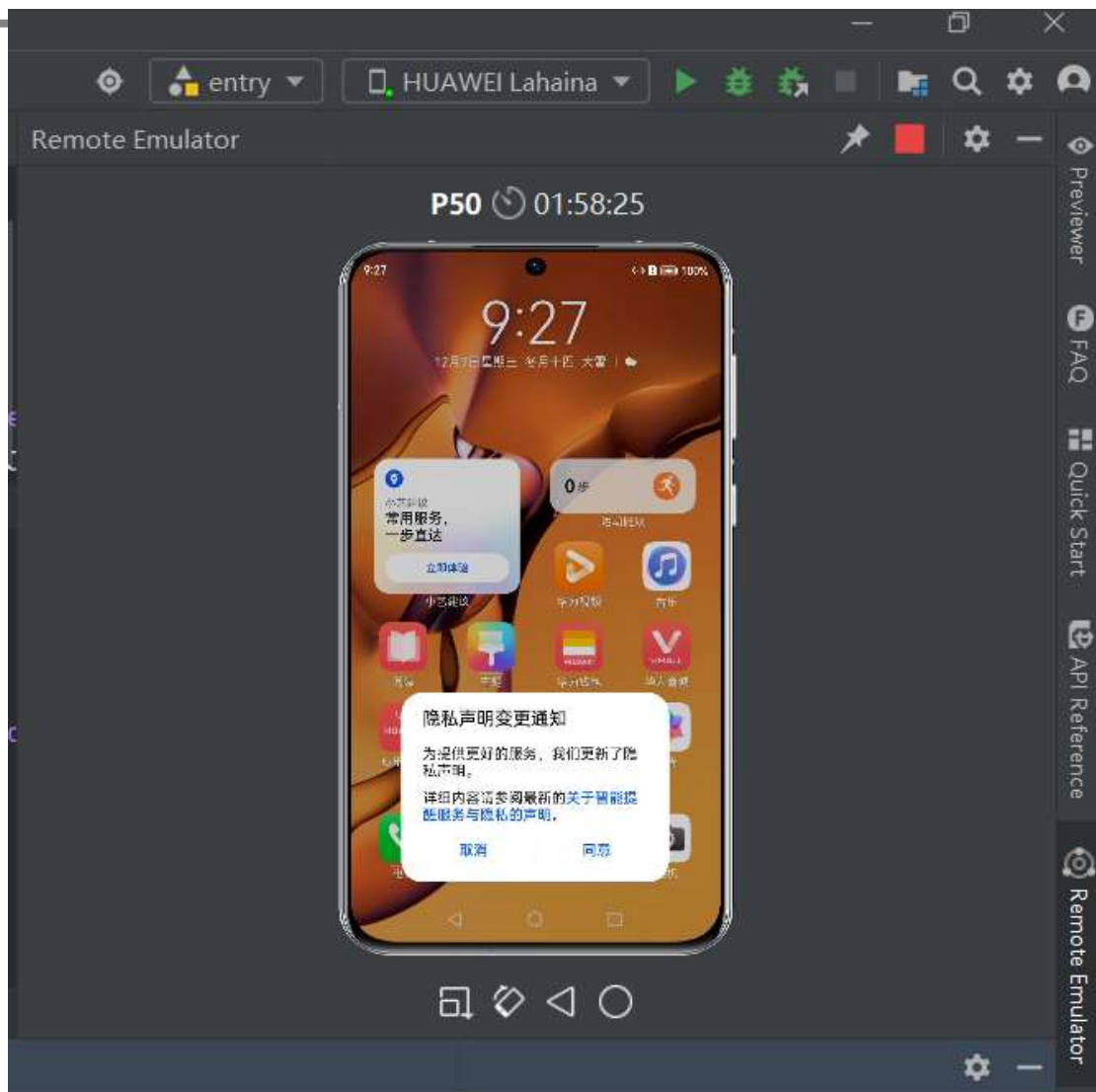
您已成功登录HUAWEI DevEco Studio客户端，请返回DevEco Studio进行下一步操作。



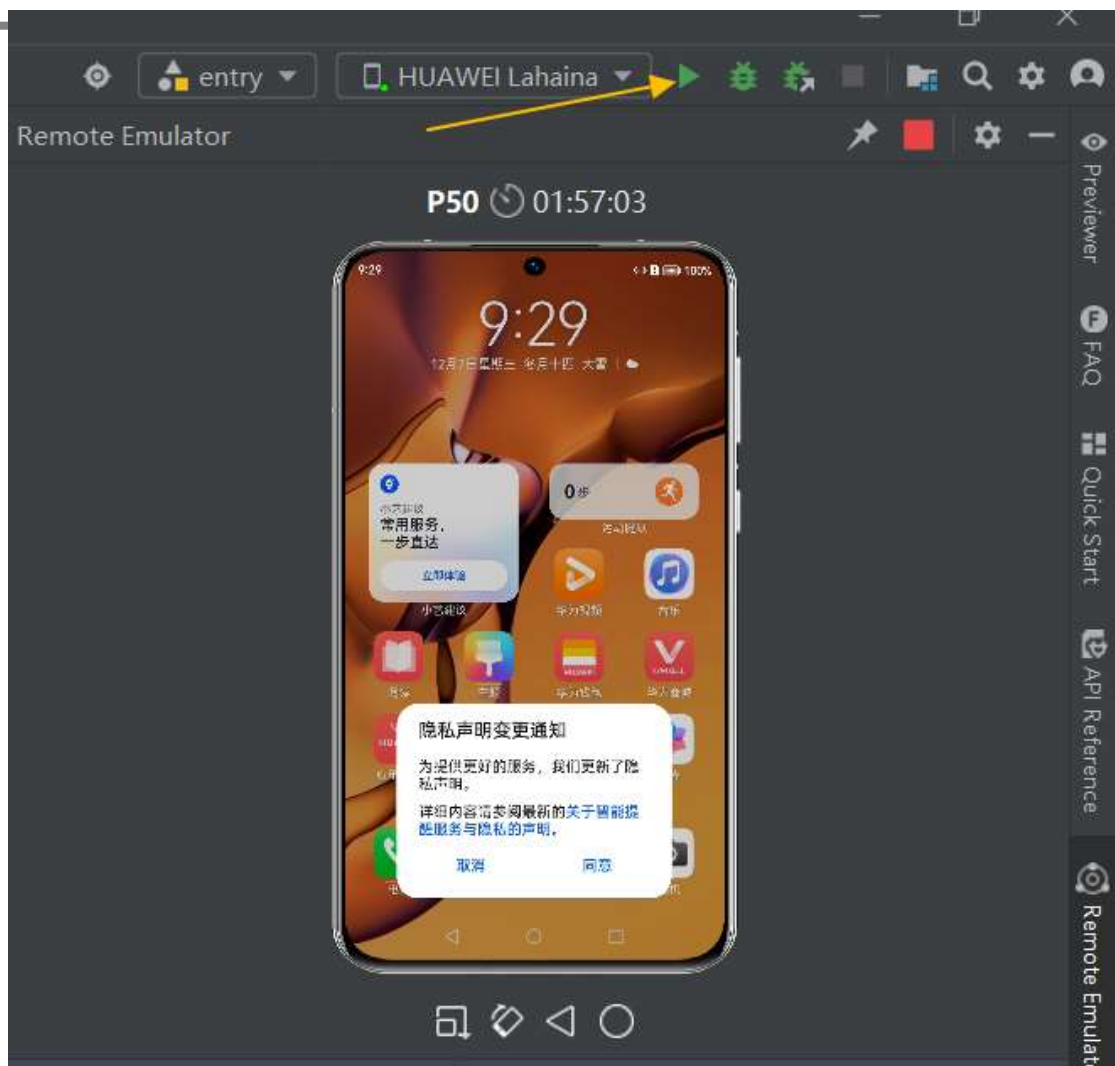
从设备列表中选择设备，注意选取 **api9** 的设备



远程模拟器启动成功（P50）：



在远程模拟器上启动项目：



程序运行成功：



点击点名按钮后效果：

