

基于 Android 平台的移动应用开发技术研究

熊国平 徐浙君

浙江邮电职业技术学院, 中国·浙江 绍兴 312366

【摘要】移动应用开发技术已经在移动互联网行业领域流行, 基于 Android 系统的应用也层出不穷。本文主要针对 Android 平台的移动应用开发技术进行研究, 分别从原生开发、跨平台开发技术层面进行研究, 对比各个开发技术的优缺点, 为从事及研究移动应用开发技术的人员提供技术参考。

【关键词】移动应用开发; Android 平台; 原生开发; 跨平台开发

【基金项目】浙江省教学改革研究项目, 编号: jg20180873。

引言

5G 技术促进移动互联网的通信技术不断发展, 进一步推动了智能手机设备在世界范围内的普及, 也影响着智能手机操作系统的技术不断发展。目前主流的移动应用开发平台为 Android 和 iOS, 它们的开发技术和运行机制也各有不同。基于 Android 系统的智能设备目前占据移动应用主要市场, 使用基于 Android 系统的智能设备用户占据比例也较高, 包括智能手机、平板、手环、智能手表、TV 等方面, 已经渗透到人们生活的方方面面, 智能设备内所安装的各种软件都是基于 Android 系统所开发出来的, 为此对基于 Android 系统的移动应用开发技术的研究是非常有必要的, 同时, 对从事 Android 移动应用开发的人员也是具有非常大的价值和意义。

1 Android 原生 APP 概述

1.1 Android 原生 APP 定义

Android 是基于 Linux 平台开发的手机和平板电脑的操作系统, 由 Google 公司开发的。

Android 原生 APP 是一种基于 Android 操作系统并且使用原生程序编写运行的第三方移

动应用程序, 主要采用 JAVA 或者 Kotlin 语言来开发应用程序, 所开发的应用程序主要在 Dalvik 虚拟机上运行。

1.2 原生 APP 开发技术

Android 四大组件、常用界面布局、自定义 view、数据存储、网络编程、动画、事件机制等方面是基于 Android 系统的主要移动应用开发技术, 本文主要介绍 Android 四大组件、常用界面布局、数据存储。

1.2.1 Android 四大组件

Android 四大组件为 Activity、Service、Broadcast Receiver、Content Provider。

● Activity: 用来显示界面和处理界面上一些控件的事件, 负责与用户交互的组件。

● Service: 没有用户界面, 在后台完成用户指定操作的服务。

● Broadcast Receiver: 广播接收者, 采用观察者模式的机制来实现消息通信。

● Content Provider: 主要用于应用之间的数据共享。

1.2.2 常用界面布局

FrameLayout (帧布局)、LinearLayout (线性布局)、RelativeLayout (相对布局)、TableLayout (表格布局)、ConstraintLayout (约束布局) 是 Android 系统的五种常用界面

布局。

● FrameLayout (帧布局): 界面布局以层级方式进行布局, 每一层的帧会包含该层的子控件, 同时, 这些帧会一个一个叠加在一起, 后加入的控件会叠加在上一个控件上层。

● LinearLayout (线性布局): 采用水平或者垂直的方式进行排列布局。

● RelativeLayout (相对布局): 采用相对定位的方式来对子控件进行布局。

● TableLayout (表格布局): 采用行、列的形式来管理控件。

● ConstraintLayout (约束布局): 主要应用以可视化的方式编写界面布局。

1.2.3 数据存储

SharedPreferences、文件存储、SQLite 数据库、ContentProvider、网络存储数据为 Android 平台的五种数据存储方式。

● SharedPreferences: 采用键值对的形式来存储数据, 数据存储在 XML 文件中, 一般用于存储状态信息。

● 文件存储: 提供了 openFileInput() 和 openFileOutput() 方法来读取设备上的文件。

● SQLite 数据库: Android 系统对应的数据库, 其使用的语法与 SQL 基本一致。

● ContentProvider: 用于存储应用程序之间的数据。

● 网络存储: 借助网络存储空间将数据存储到服务器上及获取数据信息。

2 跨平台概述

2.1 跨平台定义

跨平台是指编写出来的一套代码可以运作在不同操作系统上, 也称其具有平台无关性。“一次编写, 多平台共享代码”是程序员追求的目标, 也是减少开发费用及提高代码版本迭代的根本方法。

2.2 跨平台技术

React Native、Weex、Flutter 是目前 Android 平台跨平台技术主流的开发技术, 以下分别对这些技术进行分析。

2.2.1 React Native

React Native 利用 JavaScript 来调用 Native 端的组件来实现功能, 界面采用 JavaScript 来构建且界面展示是在运行时通过翻译成原生组件来实现的, 实际上所有界面都是用原生组件, 支持跨平台, 性能和用户体验比 Webview 效果好, 接近原生应用, 是目前许多企业的热门选择, 其优缺点如表 1 所示。

表1 React Native优缺点

优点	缺点
复用了 React 的思想, 有利于前端开发者开发	针对不同的平台提供不同的代码来开发
组件化开发, 复用率高	性能比原生差
利用 JavaScript 动态更新的特性进行快速迭代	开发技术要求高
相比于原生开发, 开发速度更快	学习曲线偏高, 文档不够完整

2.2.2 Flutter

Flutter 是一款由 Google 公司采用 Dart 作为开发语言的移动 UI 框架, 其重写了一个平台无关的渲染引擎, 提供了画布, UI 组件及渲染逻辑都在引擎上处理, 在各平台可以保持一致性, 用户体验较好, 其优劣势如表 2 所示。

表2 Flutter优缺点

优点	缺点
使用一套开发代码	前期bug较多
灵活性强, 一切都是widget	代码可读性不佳
处理动画能力强	上手较难

2.2.3 Weex

Weex 是一款采用 Web 开发技术来开发高性能原生应用的框架。其渲染引擎与 DSL 语法层是分开的, Weex 不强依赖任何特定的前端框架。Weex 页面开发所使用的前端框架目前主要为 Vue.js 和 Rax 框架, 同时 Weex 也支持这两个前端框架。Weex 的主要作用是将 Web 开发技术和原生开发技术相结合, 实现运

行性能和开发效率的高度统一, 其优缺点如表 3 所示。

表3 Weex优缺点

优点	缺点
轻量、体积小	资料少、社区活跃度不够
学习成本低	成熟的组件较少
跨平台、支持web端	坑相对较多

3 总结

通过对 Android 平台的移动应用开发技术进行研究, 分别从原生开发、跨平台开发技术层面进行研究, 对比各个开发技术的优缺点, 为 Android 开发者在技术选型及开发框架上提供了有利的参考。

参考文献:

- [1] 徐向艺, 赵郑营. Android 与 iOS 移动操作系统技术创新模式对比研究[J]. 电脑编程技巧与维护, 2020(12).
- [2] 李晓然. 智能手机操作系统—以 Android 与 iOS 对比[J]. 经济师, 2020(02).
- [3] 金诚. 移动应用跨平台开发框架的比较分析[J]. 民营科技, 2018(10).
- [4] 黑马程序员. Android 移动应用基础教程 (Android studio)(第 2 版) [M]. 北京: 中国铁道出版社, 2019.

作者简介:

熊国平 (1984-), 男, 浙江邮电职业技术学院, 硕士, 研究方向: 移动应用开发, 高性能计算。

徐浙君 (1980-), 男, 浙江邮电职业技术学院, 硕士, 研究方向: 数据挖掘、云计算。