**长春大学**

**毕 业 设 计（论 文）**

**论文题目 基于Android的移动开发**

**学习系统的设计与实现**

**专 业 计算机科学与技术**

**指导教师 杨小英**

**姓 名 徐元强**

**2019年12月**

# 摘要

近些年来，随着网络科技的不断发展，4G技术已走上成熟，5G技术成为了国内外发展的新趋势。同时，手机的发展也取得了巨大的进步，移动通讯设备与移动互联网络的结合，给人们的生活带来了无限的便利，智能手机成为了人们生活的必需品。科技在发展，知识在更新，人们只有不断的学习，才能跟得上时代的进步。传统的学习方式受到时间和地点的限制，人们对学习的需求无法得到满足，移动学习这种新型的学习方式受到越来越多的关注。本论文研究的是基于Android的移动开发学习系统的设计与实现，目的是让移动开发学习者们，合理利用移动网络的优势，随时随地不受时间与空间的限制学习。因此，研究此课题在理论上和实际生活上具有重大意义。

本文首先调研了基于Android的移动开发学习系统的研究背景及意义。其次，介绍了移动学习系统的相关技术，包括Android系统的概述和用到的第三方框架。分析了移动开发学习系统的需求，包括可行性分析、功能性分析、非功能性分析。然后重点在于移动开发学习系统的概要设计，对移动开发学习系统进行整体架构设计，系统功能模块设计。阐述了各个功能模块的详细设计。根据详细设计完成移动开发学习系统各个功能的具体实现，并利用模拟器与移动设备进行测试与修改。最后，结合本课题的研究开发工作进行总结。

该移动开发学习系统的设计与开发，主要是为移动开发学习者们提供一种辅助手段，方便移动开发学习者快速查阅知识，快速接触新知识，帮助学习者进行知识分类与整理，让学习更加高效。

**关键词**：移动学习；移动开发；Android；

**目录**

[第一章 绪论 1](#_Toc20087320)

[1.1 研究背景及意义 1](#_Toc20087321)

[1.2 国内外研究现状 2](#_Toc20087322)

[1.2.1 国外移动学习研究现状 2](#_Toc20087323)

[1.2.2 国内移动学习研究现状 3](#_Toc20087324)

[第二章 相关技术介绍 3](#_Toc20087325)

[2.1 Android操作系统概述 3](#_Toc20087326)

[2.1.1 Android操作系统的简介 3](#_Toc20087327)

[2.1.2 Android操作系统的架构 4](#_Toc20087328)

[2.2 Android数据通信技术 5](#_Toc20087329)

[2.2.1 Retrofit框架简介 5](#_Toc20087330)

[2.2.2 JSON技术 6](#_Toc20087331)

[第三章 移动开发学习系统的需求分析 6](#_Toc20087332)

[3.1 可行性分析 6](#_Toc20087333)

[3.1.1 操作可行性 7](#_Toc20087334)

[3.1.2 技术可行性 7](#_Toc20087335)

[3.1.3 经济可行性 7](#_Toc20087336)

[3.2系统总体需求分析 7](#_Toc20087337)

[3.3 系统功能性需求分析 8](#_Toc20087338)

[3.3.1 在线学习模块功能需求分析 9](#_Toc20087339)

[3.3.2 学习导航模块功能需求分析 9](#_Toc20087340)

[3.3.3 知识体系模块功能需求分析 10](#_Toc20087341)

[3.3.4 公众号模块功能需求分析 10](#_Toc20087342)

[3.3.5 项目模块功能需求分析 11](#_Toc20087343)

[3.3.6 个人中心模块功能需求分析 11](#_Toc20087344)

[3.4 系统非功能性需求分析 12](#_Toc20087345)

[3.4.1 屏幕适配 12](#_Toc20087346)

[3.4.2 手机续航 12](#_Toc20087347)

[3.4.3 流畅性 13](#_Toc20087348)

[3.4.4 内存管理 13](#_Toc20087349)

[第四章 移动开发学习系统的概要设计 13](#_Toc20087350)

[4.1 系统架构 13](#_Toc20087351)

[4.1.1 移动开发学习系统架构设计 13](#_Toc20087352)

[4.1.2 移动开发学习系统框架设计 14](#_Toc20087353)

[4.2 系统功能模块设计 15](#_Toc20087354)

[4.2.1在线学习功能模块设计 15](#_Toc20087355)

[4.2.2 学习导航功能模块设计 16](#_Toc20087356)

[4.2.3 知识体系功能模块设计 16](#_Toc20087357)

[4.2.4 公众号功能模块设计 17](#_Toc20087358)

[4.2.5 项目功能模块设计 17](#_Toc20087359)

[4.2.6 个人中心功能模块设计 18](#_Toc20087360)

[第五章 移动开发学习系统的详细设计 19](#_Toc20087361)

[5.1 开发环境 19](#_Toc20087362)

[5.1.1 系统开发环境 19](#_Toc20087363)

[5.1.2 Android客户端开发环境 19](#_Toc20087364)

[5.2 功能模块详细设计与实现 20](#_Toc20087365)

[5.2.1 在线学习功能模块 20](#_Toc20087366)

[5.2.2 学习导航功能模块 22](#_Toc20087367)

[5.2.3 知识体系功能模块 24](#_Toc20087368)

[5.2.4 公众号功能模块 26](#_Toc20087369)

[5.2.5 项目功能模块 28](#_Toc20087370)

[5.2.6 个人中心功能模块 30](#_Toc20087371)

[第六章 移动开发学习系统的测试 31](#_Toc20087372)

[6.1 功能测试 32](#_Toc20087373)

[6.2 性能测试 33](#_Toc20087374)

[第七章 总结 33](#_Toc20087375)

[附录：源程序清单 35](#_Toc20087376)

[参考文献 148](#_Toc20087377)

# 第一章 绪论

## 1.1 研究背景及意义

随着移动网络技术和通讯设备的快速发展，人们的生活方式以及学习方式都发生了翻天覆地的变化。人们对知识的渴望，传统的学习方式已经无法满足，近年来，人们获取知识的途径也不断地随之改变。移动学习作为一种新的学习方式出现在人们的视野中，移动学习是一种在移动设备帮助下的能够在任何时间、任何地点发生的学习[[[1]](#endnote-2)]。北京大学现代教育技术中心移动教育实验室对移动学习的定义是，移动学习是指学习者和教育者使用手机、平板电脑等智能移动终端，依靠更成熟的无线移动网络、互联网和多媒体技术，灵活地开展互动教学活动[[[2]](#endnote-3)]。

目前，我国的教育教学仍以初等教育、中等教育和高等教育为主。学习者获取知识的主要途径是通过以上教学方式，而对于毕业人士，职场人士，已经没有大量的时间去课堂集中学习。然而，随着科技的快速发展，新知识层出不穷，知识更新周期不断缩短，传统的教学方式无法满足学生对知识获取的需求，课堂教学也无法适应这个时代的新需要。移动学习方式的出现，打破了传统教学的束缚，在这种学习模式下，不仅能满足在校学生的学习需要，还能满足爱好学习者与职场人士的学习需要。

根据中国互联网络信息中心发布第43次《中国互联网络发展状况统计报告》。报告显示，截至2018年12月，我国网民规模达8.29亿，全年新增网民5653万，互联网普及率为59.6%，较2017年增长3.8%，我国手机网民规模达8.17亿，占网民的98.6%，我国在线教育用户规模达2.01亿人，较2017年增长29.7%;其中手机端在线教育用户规模达 1.94亿人，较2017年增长63.3%[[[3]](#endnote-4)]。可以得知，手机设备已经成为人们上网的主要工具，并且随着生活水平不断的提高，手机用户将会不断增长，在线教育的用户规模也会逐渐扩大。在这种大的背景下，移动学习的发展已经成为一种新的趋势，教育资源和教育服务通过网络技术合理利用，能够带给学习者新的学习感受。在移动互联网环境下，开发一个能够满足学习者学习需求的学习系统十分有前景。

本论文依靠玩Android学习网站开发一种符合移动开发学习者学习要求的学习系统。移动开发学习资源与移动学习两者互相结合，学习者安排学习时间、地点更加自由，并且能够提高学习者学习效率。随着互联网时代的到来，人们越来越迫切不断接触新知识，移动学习方式的出现已成为了一种新思路。因此，移动学习方式以一种新的学习方式出现，将会吸引越来越多的学习者。并且能够真正意义上的实现自主学习，最终实现教育社会化与社会教育化。

## 1.2 国内外研究现状

### 1.2.1 国外移动学习研究现状

目前，移动学习引起了全球世界各地的注意，更多的学者们意识到了它的潜能。现在还没有形成完整的理论体系，但国外的学者们依然对此有独特的见解。“移动学习”这一概念，最早是由国外提出，关于移动学习的相关研究相比于国内较早，国外关于移动学习的项目案例也是层出不穷。

(1)Handheld Andrew 项目

1994年，卡内基梅隆大学(Carnegie Mellon University)开始了一个叫 Wireless Andrew 移动项目的研究，是对移动学习最早开始探索性研究的一个初步案例。卡内基梅隆大学开发了一些基于PDA的应用软件对教育进行评价，参与者可以借助无线通信技术在校园里随时随地的使用该软件[[[4]](#endnote-5)]。

(2)移动创新项目

开始于2010年秋季的移动创新项目是美国博伊西州立大学(Boise State University)多个项目的整合，主要包含大学移动图书馆、图书馆电子学习项目和移动学习“m- Learning”学者计划。该大型项目要求发展一系列以信息技术为依托的跨校园合作计划[[[5]](#endnote-6)]。

(3)开放式的网路课程

Coursera、Udacity、MOOC 都是目前活跃的非营利网络在线学习平台。它们依托全球100多所的知名大学（如哈佛大学、麻省理工学院、加州大学伯里克分校等）的网络课程资源，为学习者提供开放的学习平台。它们的课程包罗万象，有物理、化学、计算机科学、统计学、生物、地理等。学习者只要在这些学习平台上注册账号后，就可以通过网络收看课程视频[[[6]](#endnote-7)]。

### 1.2.2 国内移动学习研究现状

国内研究移动学习起步比较晚，而且与国外的研究水平存在一些差距。在2000年上海电大举办的建校40周年学术会议时，远程教育家Desmond Keegan提出《从远程学习到电子学习再到移动学习》的报告，首次将移动学习概念介绍到中国，标志着移动学习开始引入我国[[[7]](#endnote-8)]。然后，我国才真正开始研究移动学习，并且创办很多具有代表性和实践意义的项目。

(1)“移动教育”理论与实践项目

2001年11月，以“移动教育”为主题的项目由我国教育部高教司研究提出，并由北京大学现代教育中心移动教育实验室着手研究，迅速成立了理论知识研究与应用实践项目的研究小组。这是国内第一个有关移动学习方面研究的实验室。2005年12月项目结束。该项目制定了教学资源的准则与开发规范，提高了教学服务平台的智能性和提供多功能的教育服务。

(2)中国无线大学计划

2006年，中国教育和科研计算机网(CERNET)与英特尔推出了“中国无线大学计划”。该计划打破了传统教学模式的局限性，移动学习平台在中国各大高校间统一建立，实现高校间的资源共享与教学分享。从而促进了全国教育资源的整合，并为全国学生提高更加高效的学习方式。

(3)“互联网+教育”

2012年提出“互联网+”这一概念一直受到各界关注，2015年初《政府工作报告》中首次提出“互联网+计划”，寓意着“互联网+”开始已经到了国家战略层面的高度。“互联网+教育”促进了我国教育走向新的阶段，共同探索互联网与教育的深度融合。它是对传统教育方式的一种升级，使用一个教育专用网、一部移动终端设备，学生可以随时随地，任意选择学习资源的一种新学习模式。

# 第二章 相关技术介绍

## 2.1 Android操作系统概述

### 2.1.1 Android操作系统的简介

Android操作系统是一个基于Linux内核的嵌入式操作系统，安装在智能手机、平板电脑等智能移动终端上，用于为移动设备管理硬件和软件资源[[[8]](#endnote-9)]，主要使其作用相当于个人电脑安装上Windows操作系统。Android操作系统最初由Andy Rubin开发，在2005年8月由Google收购。2007年11月，Google与多家硬件制造商、软件开发商及电信运营商联盟共同研发改良Android。2008年10月，HTC公司发布了世界上第一台基于Android操作系统的手机。随着这部手机的成功发布，Android操作系统的使用迅速扩大到其他领域上。2011年，Android操作系统市场份额跃升全球第一。

### 2.1.2 Android操作系统的架构

Android操作系统采用分层的架构，架构如图2-1所示。

从架构图可知，Android分为四个层，从高到低分别是应用程序层、应用程序框架层、系统运行库层和Linux内核层[[[9]](#endnote-10)]。下面分别简单介绍和分析这四层的运行工作。

（1）Linux内核层

Android平台的基础是Linux内核，Linux内核为Android提供了许多关键性的功能。例如，能够删除内核中不安全或不可靠的部分，能够实现安全的IPC进程间通信机制等。

（2）系统运行库层

以C和C++编写的原生库，为许多核心Android系统服务和组件构建原生代

码提供了支持。还可以通过Android框架访问OpenGL ES，以支持应用程序中操作和绘制2D、3D图形。

这一层还包括Android Runtime库，在Android系统5.0之前，Android Runtime是Dalvik，应用程序每次运行时，它使用即时编译器将字节码转化为机器码。在Android系统5.0之后，Android Runtime是ART，它是进行一次预编译将字节码转化为机器码并存储在本地，应用程序每次运行时不会执行多次编译。

（3）应用框架层

这一层是让Android开发者可以通过以Java语言编写的API使用Android

系统内的功能集合。比如，用于控制整个应用生命周期的Activity管理器，在程序状态栏显示自定义提醒的通知管理器，用于访问非代码资源的资源管理器等。

（4）应用层

Android系统随附的系统应用和用户自己选择安装的用户应用都属于这一层。系统应用既可以作为用户应用，又可以为开发者提供从其应用需要访问的功能。



图2-1 Android系统架构图

## 2.2 Android数据通信技术

### 2.2.1 Retrofit框架简介

Retrofit是Square公司开发的一款针对Android网络请求的框架，Retrofit实际上算不上一个新的框架，它是对OkHttp框架的一个封装[[[10]](#endnote-11)]。与其他网络框架不同的是，它更多使用运行时注解的方式提供功能[[[11]](#endnote-12)]。它的基本用法如下。

1. 首先配置Android工程目录下build.gradle文件，并添加依赖包，在

manifest文件中加入访问网络的权限。

1. 编写网络请求接口，标明请求方式注解，输入需要访问的地址。
2. 创建Retrofit对象，并创建接口文件，接下来通过Call对象请求网络。

使用Retrofit框架具有诸多优点，如支持同步、异步和RxJava；源代码设

计中运用了大量的设计模式进行解耦；使用非常方便、灵活；因此，合理的利用Retrofit网络请求框架可以提高开发效率，也让应用程序的网络请求模块具有一定的健壮性。

### 2.2.2 JSON技术

本系统实现的功能主要是通过Android移动端与服务端数据交互来完成，因此，选择一种快速、安全、通用、数据通信量小的数据通信方式对移动设备来说是非常重要的。本系统采用了JSON技术作为数据交互方式，JSON的全称是JavaScript Object Notation，是一种轻量级的数据交换格式。JSON数据格式相比于XML比较简单轻便，方便开发人员读写，适合机器本身生成和解析。

JSON 支持Java、C、C#、PHP 等多种服务端语言，服务器端解析方便，它也提供这些语言之间相互进行数据交换的方式，JSON采用两种结构[[[12]](#endnote-13)]：

（1）“名称/值”对的集合，“名称/值”可以表示对象的属性，对象其实就是属性的集合，所以这个结构可以用来表示对象。

（2） 数组结构，表示对象的集合，JSON 能够将多个不同的对象组合在一起，形成一个值的有序列表，易于阅读和便于机器生成。

JSON具有数据格式简单，比较方便读写，利用压缩格式，占用带宽也比较小等多种优点。同时，JSON完全是一个JavaScript对象,不需要创建DOM对象，服务器端可以跨域传输信息也可以直接使用其格式。

# 第三章 移动开发学习系统的需求分析

## 3.1 可行性分析

可行性分析是通过对项目的市场需求、建设规模、资金筹措等主要内容进行研究。从技术、工程、经济等方面进行调查研究和分析比较，从而确定此系统是否有必要实现。下面从操作可行性、技术可行性和经济可行性三个方面进行分析。

### 3.1.1 操作可行性

现在我们国家的4G网络基本上做到了全面覆盖，而且5G网也在进行紧锣密鼓的测试，而且随着智慧城市的不断发展与蔓延，很多公共场所像学校、医院、餐厅、商场以及企业办公大楼都实现了WIFI的全面覆盖[[[13]](#endnote-14)]。网络的普及解决了移动学习对网络数据传输性能的要求，学习者可以充分利用这些便利条件，随时随地利用碎片时间登录移动开发学习系统进行学习。

### 3.1.2 技术可行性

从技术角度考虑，实现移动开发学习系统达到学习者随时随地学习的目的是可行的。本系统采用的是C/S架构，利用Java语言使用Android Studio开发工具进行软件开发。客户端使用简单快速的SQLite数据库存储用户搜索记录。服务器端采用玩Android网站开放API提供数据，在客户端与服务器端之间采用JSON格式的数据进行传输。在硬件方面上，基于Android的移动开发学习系统允许在Linux、Windows、UNIX等系统的PC机上进行开发。调试程序时可以使用自己的手机调试，也可用Android Studio自带的模拟器调试。可见，系统的开发平台比较成熟可行。

### 3.1.3 经济可行性

本系统采用的Android Studio开发工具在Android官方网站可以免费下载，官方网站提供的开发文档、示例代码和设计指南也都可以进行参考，这为开发人员开发Android应用程序节省了大量的时间成本。Android系不仅免费且开源，现在正在以非常快速度持续增长，目前市场份额占据第一。在开发系统过程中除了投入过多的精力外，不涉及任何的经济费用。因此，综合以上因素，基于Android的移动开发学习系统，从经济角度考虑，完全可行。

## 3.2系统总体需求分析

基于Android的移动开发学习系统运用Client/Server架构交互。客户端基于Android系统开发，移动开发学习系统的客户端部署在用户的移动终端设备上，客户端用户通过移动终端设备，连接网络在线访问服务器上的资源进行学习。移动学习平台的服务器端，采用玩Android网站开源接口。移动客户端用户使用移动开发学习系统，向Web服务器发送数据请求，当服务器收到用户请求后，会向客户端发送对应请求的JSON数据，最后客户端将JSON数据进行处理并展示给移动客户端用户。

本文设计的移动开发学习系统主要服务于想要学好Android开发的学习人员，适用于在校的学生，工作的职场人士和自学人员。当学习人员进入移动开发学习系统后，要求学员能够查阅移动开发学习人员发布的原创技术文章，能够访问学习时常用的网站，知名开发者个人博客等。客户端还要支持知名微信公众号展示、知识分类展示、完整项目学习，并且对自己感兴趣的内容进行标记等。

综合上述移动开发学习用户对系统的功能需求，可以确定基于Android的移动开发学习系统的设计与实现的功能至少包括：在线学习、学习导航、知识体系管理、公众号管理、项目管理、个人中心管理、用户搜索、用户收藏等。客户端顶层用例图，如图3-1所示。

学员

图3-1移动开发学习系统顶层用例图

## 3.3 系统功能性需求分析

### 3.3.1 在线学习模块功能需求分析

系统首页模块主要负责向移动开发学习人员展示移动开发学习者在各大博客平台发布的原创文章或技术分享，首页需要展示的文章分为四大类：推荐文章、置顶文章、最新文章和历史文章。用户能够对学习感兴趣的文章添加收藏，只有在用户登录账号的条件下才可以收藏，否则提示用户登录。同时首页提供用户对自己想要学习的内容进行关键字搜索功能，热门关键词也有必要显示，用于告知用户当下最流行的关键技术。首页模块功能用例图，如图3-2所示。

学员

图3-2 在线学习模块功能用例图

### 3.3.2 学习导航模块功能需求分析

学习导航功能模块主要是对学习者学习时起到指引作用，当学习者学习Android开发知识不知从何学起时，移动开发学习系统提供常用学习网站，在线学习网站，知名开发者、公司博客汇总等功能，给学习者一个学习方向。针对于移动学习开发者，总结日常开发常用工具并整理，以便下次查找。学习导航不仅提供知识导航，还提供互联网资讯导航、求职招聘导航等等，是一个综合性比较强的模块。学习导航模块功能用例图，如图3-3 所示。

学员

图3-3 学习导航模块功能用例图

### 3.3.3 知识体系模块功能需求分析

由于大多数学习者日常学习到的都是一些零碎的、分散的相对独立的知识概念或观点，导致学习者自身的知识点或者经验都是碎片化，不能形成体系。所以移动开发学习系统需要提供知识体系模块，将碎片化的知识进行收集、梳理和归纳，提炼出各个知识的主要模块，形成一定联系的知识系统。比如Android四大组件、常用控件、开发环境、基础知识等多类模块。知识体系模块功能用例图如图3-4。

学员

图3-4 知识体系模块功能用例图

### 3.3.4 公众号模块功能需求分析

公众号是移动互联网环境下出现的一种新的信息传播方式，通过公众号可以向微信给用户推送新闻信息、娱乐资讯、个人原创等。由于微信用户可能订阅多个公众号，并且这些公众号没有进行分类整理，需要用户经过选择后才进行查看。系统中提供公众号功能模块主要负责对公众号管理功能，将多个知名Android开发者公众号和知名公司团队微信公众号进行统一管理，并且学员可以对感兴趣的公众号内容进行收藏，知名开发者公众号有鸿洋、郭霖、玉刚说等，知名公司团队公众号有美团技术团队、谷歌开发者、Android达摩院等。公众号功能模块用例图如图3-5。

学员

图3-5 公众号模块功能用例图

### 3.3.5 项目模块功能需求分析

项目功能模块是为了让学习者将所学到的知识进行综合运用。通过在线学习模块学习到的内容是琐碎的知识点，即使有知识体系模块帮助我们构建自己的知识体系，但也都是理论上的知识。因此，项目模块可以帮助学习者从理论知识过渡到实际应用，这会加深对所学知识的深入理解，并且便于学习者记忆。学习者进入到项目模块时，可查看多种不同类型的项目，对于想要学习的内容查看详情，也可以分享给其他移动开发爱好者。项目功能模块用例图如图3-6。

学员

图3-6 项目模块功能用例图

### 3.3.6 个人中心模块功能需求分析

个人中心功能模块主要负责学习者个人账号管理、学习资源管理。学习者应当向服务器注册个人用户信息，只有通过移动开发学习系统成功注册的用户，才能够登录系统，才能授权使用系统部分功能。用户登录需要输入用户名和密码，系统会将用户输入的用户名和密码与服务器中心存储的信息进行验证，如果通过验证，允许用户登录系统，允许用户查看个人收藏等信息。关于模块是对系统的简介以及意见反馈。个人中心功能模块用例图如图3-7。

学员

图3-7 个人中心功能模块用例图

## 3.4 系统非功能性需求分析

### 3.4.1 屏幕适配

众所周知，由于Android系统的开放，任何用户和硬件厂商都可以对Android系统进行定制，所以出现Android碎片化的概念，碎片化是指市面上出现Android版本众多，Android设备尺寸众多和分辨率众多的现象。在日常开发应用程序中，开发人员要根据设备的不同分辨率和不同尺寸设置不同的资源。另外，对于界面的布局容器而言，Android系统提供给开发者可以使用多种不同布局的选择，虽然看起来比较灵活，但是开发者也需要慎重选择使用某种布局。有时不合理的使用布局会影响界面的显示效果，因此，合理的选择布局对界面的设计也起到至关重要的作用。

### 3.4.2 手机续航

在智能手机发展过程中，手机续航能力是各大手机生产商一直在突破的问题，一部质量过关的手机，在电池容量一定的情况下，耗电能力应该较小，以保证手机的续航能力。手机的续航能力与手机自身硬件配置有关，比如智能手机的屏幕越来越大，手机的内置传感器越来越多。也与手机上安装的软件有关，随着手机上安装的APP越来越多，每个应用程序的功能也丰富多彩，这些都对电池续航能力提出了巨大挑战。所以，在开发手机应用程序时，软件产品的功耗问题需要重点考虑。

### 3.4.3 流畅性

Android系统每隔16ms会发出VSYNC信号重新绘制一次Activity，触发对UI的渲染机制，如果16ms内没有完成绘制就会造成卡顿现象。用户在使用APP时，如果经常发现某个应用程序出现卡顿、程序无响应、自动退出等异常现象，必定会造成用户的疑惑和反感，用户甚至直接将其进行卸载，不会在使用，因此，开发人员在开发应用程序时，需要不断的测试以及重度使用检测系统的流畅性。

### 3.4.4 内存管理

内存管理的目的就是让我们在开发手机应用程序过程中如何有效的避免应用出现内存泄漏的问题。内存泄漏简单的讲就是，程序中该释放的对象没有及时释放，一直被某个或某些实例所持有却不再被使用导致GC不能回收。众所周知，。Android设备作为一种移动设备，不管是内存还是CPU的性能都受到一定的限制，这也意味着Android程序不可能无限制的使用内存和CPU资源[[[14]](#endnote-15)],如果手机内存被应用程序占据大量的空间，并且垃圾回收机制不能回收应该回收的对象，应用程序会变卡，同时手机也会变卡，从而导致用户体验变差。需要特别注意的是，当程序中加载大量图片时，必须要考虑到程序内存溢出情况。当图片不在使用时，需要对图片及时回收并释放内存。

# 第四章 移动开发学习系统的概要设计

## 4.1 系统架构

### 4.1.1 移动开发学习系统架构设计

移动开发学习系统主要根据移动开发学习者日常学习习惯以及解决学习难点来使用，要充分考虑到移动学习的特点和客户需求，设计系统总体架构时，本系统采用C/S架构设计,利用软件四层架构模型进行开发。其中，用户层表明系统的用户组成；表示层主要负责接收数据服务层传输的用户请求数据以及经过业务逻辑层处理过后的数据展示；业务逻辑层主要是对数据服务层的数据处理，它处于表示层和数据服务层的中间层次，具有一定的桥梁作用；数据服务层主要负责对数据库中原始数据的处理，为其他三层提供数据服务。

移动开发学习系统使用这四层结构模型，在架构上层次清晰，每一层都可以独立测试，其他层的接口通过模拟解决，充分体现出软件开发过程中高内聚、低耦合的思想。同时，让复杂的系统简单化，在系统开发方面上，实现了界面与代码的分离，有利于降低层次之间的依赖性，有利于提高系统开发效率。总体架构图如图4-1。

学习者

用户层

移动终端设备

表示层

在线学习模块

学习导航模块

公众号模块

知识体系模块

个人中心模块

项目模块

业务逻辑层

数据服务层

服务器

图4-1系统总体架构图

### 4.1.2 移动开发学习系统框架设计

基于Android的移动开发学习系统的功能结构如图4-2所示。当学习者进入移动开发学习系统后，将会看到最新的、推荐的在线学习资源。学习导航是一个综合性比较强的模块，学习者不仅可以选择自己想要的学习方式，还可以找到自己想要的开发资源。通过知识体系可查看关于Android的开发知识分类，利用公众号随时查看各知名开发者个人技术分享，合理使用项目将自己所学过的知识进行运用。个人中心管理自己的学习资源，修改个人资料等。

其中在线学习模块是移动开发学习系统的最核心功能，包括推荐文章、置顶文章、最新文章、历史文章、查看文章详情、收藏文章和搜索文章；学习导航主要包括学习者经常学习时用到的常用网站、公司博客、开发社区、常用工具、互联网资讯和在线学习等多个模块；知识体系模块为学习者梳理知识时提供了便利，以便于将学习者所学到的碎片化知识梳理自己的知识体系，主要的知识体系有开发环境、基础知识、四大组件、常用控件、图片加载等；个人中心模块提供维护个人信息，注册、登录、退出、查看收藏文章，关于等。

基于Android的移动开发学习系统

项目模块

公众号模块

知识体系模块

学习导航模块

个人中心模块

在线学习模块

图4-2 系统功能结构图

## 4.2 系统功能模块设计

### 4.2.1在线学习功能模块设计

在线学习功能模块的设计主要为学习者提供相关学习资源功能的服务。学习者可以暂且不登录移动开发学习系统，选择自己感兴趣的学习资源进行学习，当学习者想要收藏此学习内容时，学习者必须登录自己的账号。结合移动开发学习系统特点和用户需求分析，在线学习功能模块的设计应包括“查看推荐文章”、“查看置顶文章”、“查看最新文章”等功能。在线学习功能模块设计结构图图

4-3。

在线学习功能模块

查看最新文章

查看推荐文章

查看置顶文章

图4-3 在线学习功能模块设计结构图

### 4.2.2 学习导航功能模块设计

学习导航功能模块的设计主要为学习者起到引导作用。学习者想要学习而不知道有哪些学习资源，不知道从何学起，此模块会发挥很大作用。根据移动开发学习系统的特点和功能需求分析，在学习导航功能模块设计中，提供了“查看常用网站”、“在线学习”、“查看源码”、“开发社区”等功能。学习导航功能模块设计结构图4-4。

学习导航功能模块

在线学习

常用网站

常用工具

图4-4学习导航功能模块设计结构图

### 4.2.3 知识体系功能模块设计

知识体系功能模块是为学生梳理知识、整理和分类知识的模块。该模块就是将互联网上零碎、分散、相互独立的移动开发相关知识或相关观点分类综合在一起，使之各类知识更加模块化。目前已整理的模块有“基础知识”、“四大组件”、“用户交互”、“网络访问”等。学习者可以按类学习，对自己感兴趣的内容也可收藏。知识体系功能模块设计结构图4-5。

知识体系功能模块

练习常用控件

学习四大组件

巩固基础知识

图4-5 知识体系功能模块设计结构图

### 4.2.4 公众号功能模块设计

公众号功能模块的设计主要为学习者查看知名开发者公众号、知名公司公众号提供便利的服务。此模块将知名开发者和知名公司公众号整合在一起，其中包括鸿洋、玉刚说、谷歌开发者、其卓社等，同时支持查看各公众号内容文章详情以及学习资源收藏功能，公众号功能模块设计结构图4-6。

公众号功能模块

收藏公众号文章

查看公司公众号

查看个人公众号

图4-6 公众号功能模块设计结构图

### 4.2.5 项目功能模块设计

项目功能模块的设计主要为提高学习者知识运用能力。理论知识过多没有实践能力，不能综合运用，这是开发者的一大弊端。项目模块是对Android某些知识点或某块或多块知识的综合运用，其中包括个人完整项目、跨平台应用等。学习者还可以对学习的资源进行分享。项目功能模块设计结构图4-7。

项目功能模块

分享项目

浏览项目

查看项目详情

图4-7 项目功能模块设计结构图

### 4.2.6 个人中心功能模块设计

个人中心功能模块的设计是对登录用户进行信息验证和审核等功能处理。本文的移动开发学习系统采用用户名和密码的验证方式，无账号需要注册。只有登录成功后才可查看自己的收藏学习资源。个人中心功能模块设计结构图4-8。

个人中心功能模块

查看我的收藏

用户注册

用户登录

图4-8 个人中心功能模块设计结构图

# 第五章 移动开发学习系统的详细设计

## 5.1 开发环境

### 5.1.1 系统开发环境

基于Android的移动开发学习系统的设计与实现主要是在Window10系统环境下完成，由于本系统的客户端选择Java作为开发语言，所以在系统开发之前需要下载软件开发工具包JDK。JDK 是Java语言的软件开发工具包，它包含了Java的运行环境、工具集合、基础类库等内容[[[15]](#endnote-16)]。系统使用的JDK版本为“1.8.0\_211”，安装成功后需要对系统环境变量进行配置，只有正确配置成功后，才能正常运行开发软件。验证配置成功如图5-1所示。

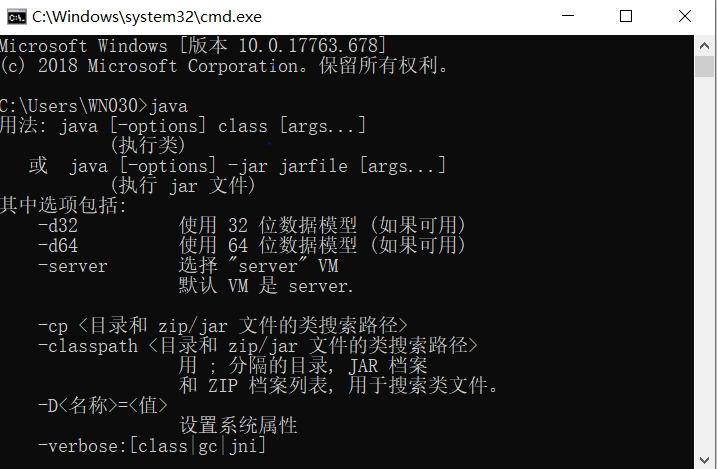


图5-1 Java环境变量配置成功图

### 5.1.2 Android客户端开发环境

本移动开发学习系统实现的Android客户端使用Android Studio开发环境，Android Studio是基于IntelliJ IDEA的适用于Android应用开发的官方集成开发环境。除了IntelliJ强大的代码编辑器和开发者工具，Android Studio还提供更多可提高Android应用编译效率的功能。

本系统使用的开发工具版本为Android Studio 3.4.1，可直接访问Android官方网站直接下载并安装。安装时需要连接网络，Android Studio会自动提示下载Android SDK以及ADT(Android Development Tools)等插件。本系统测试使用的手机是三星Galaxy A6s，Android版本为8.1.0（API 27）。

## 5.2 功能模块详细设计与实现

### 5.2.1 在线学习功能模块

在线学习功能是移动开发学习系统最主要的功能。此模块为移动开发学习者者提供学习内容，并且内容会随时更新，学习者也可以手动拉下刷新，目的是让学习者能够不断接触新知识。在线学习模块功能包含查看学习内容、搜索功能、文章收藏功能等。

查看学习内容是学习者学习移动开发知识的主要途径，在互联网环境下，每天都有成百上千篇文章发布，经过系统筛选分为推荐文章，置顶文章，最新文章和历史文章。这些不同种类的文章将会同时显示在同一页面内。

当学习者启动移动开发学习系统进入启动页，会看到“欢迎来到移动开发学习系统”提示语，启动页加载完毕后，进入在线学习功能模块并加载相对应布局。此时，Android客户端将会向服务端发送获取推荐文章的URL,当客户端成功接收到服务端发送的信息后将以轮播图的方式展示。推荐文章请求成功后紧接着继续请求学习文章接口。请求成功后的文章将会以列表的形式展示，如果客户端请求失败，提示用户做相应操作。具体流程图如图5-2所示。

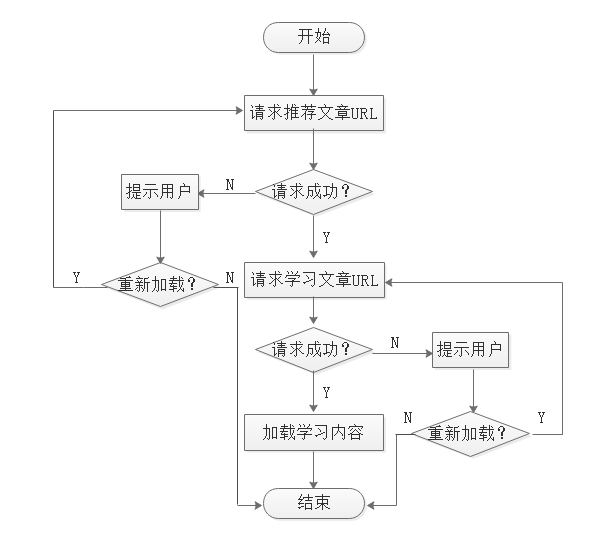


图5-2在线加载学习内容流程图

用户在进入移动开发学习系统时，首先会创建HomePageFragment类，该类首先要在资源文件夹下加载fragment\_homepage.xml布局文件，该布局内包含了RecyclerView和SmartRefreshLayout两种控件。其中SmartRefreshLayout用来支持用户上拉加载更多，下拉刷新学习内容的控件。用户上拉加载更多，下拉刷新事件关键代码如下：

private void initRefreshLayout() {  
 //下拉刷新  
 hpSmartRefreshLayout.setOnRefreshListener(new OnRefreshListener()

{  
 @Override  
 public void onRefresh(@NonNull RefreshLayout refreshLayout) {  
 presenter.refreshLayout(false);  
 refreshLayout.finishRefresh();  
 }  
 });  
 //上拉加载更多  
 hpSmartRefreshLayout.setOnLoadMoreListener(new

OnLoadMoreListener() {  
 @Override  
 public void onLoadMore(@NonNull RefreshLayout refreshLayout)

{  
 presenter.loadMore();  
 refreshLayout.finishLoadMore();  
 }  
 });  
}

在线学习功能模块实现界面如图5-3。



图5-3在线学习模块界面

### 5.2.2 学习导航功能模块

学习导航功能是辅助学习者查找学习内容的功能，当学习者不知道有哪些学习网站或者是想要学习某块移动开发技术时，可以根据学习导航整理好的模块，能够有效的帮助学习者找到想要的学习资源。

学习导航内的子模块不仅比较多，而且各个模块包含的内容也比较多，移动端的屏幕大小不能满足，所以客户端分为两列列表显示，一列显示学习导航内容整理的子模块，一列显示子模块相对应的内容。首先，Android客户端向服务器端发送学习导航数据请求，请求成功后，客户端根据服务器端返回的结果信息分别显示子模块列表、子模块内容列表，默认将选中的子模块文字置黑，未选中置灰，具体流程图如图5-4所示。

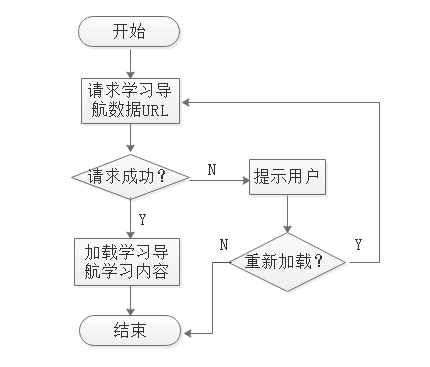


图5-4 学习导航学习流程图

当用户选择学习导航模块时，Android客户端会加载NavigationFragment类，首先该类将请求学习导航的数据分成两列列表展示。用户选择左侧列表上的学习模块时，右侧列表会自动滑动到该模块下的内容对应的位置。当用户选择右侧列表模块内容时，左侧列表会自动滑动到该内容对应学习模块的位置。用户实现左侧、右侧列表点击效果的部分关键代码如下：

rv\_navigation.addOnScrollListener(new RecyclerView.OnScrollListener() {  
 @Override  
 public void onScrollStateChanged(RecyclerView recyclerView, int

newState) {  
 super.onScrollStateChanged(recyclerView, newState);  
 if (scroll && (newState == RecyclerView.SCROLL\_STATE\_IDLE)) {  
 scrollRecyclerView();  
 }  
 rightLinkageLeft(newState); //滑动左侧学习模块列表  
 }  
  
 @Override  
 public void onScrolled(RecyclerView recyclerView, int dx, int dy)

{  
 super.onScrolled(recyclerView, dx, dy);  
 if (scroll) {  
 scrollRecyclerView(); //滑动右侧学习模块内容列表  
 }  
 }  
 });

学习导航功能模块实现界面如图5-5。



图5-5学习导航模块界面

### 5.2.3 知识体系功能模块

知识体系功能是将互联网上零散的、琐碎的Android开发相关知识进行分类整理，通过知识体系的结构提供学习者学习。学习者每天能够阅读大量的移动开发技术文章，但是记住的少之又少，原因是没有结构性的去学习，没有看一些体系化的内容。

当学习者选择知识体系模块时，移动客户端会自动向服务端发送知识体系的数据请求，客户端成功接收到数据后，将知识体系分类后的知识模块按照列表的形式展示给学习者，当学习者选择某一知识模块，系统自动加载该模块下知识点的详细内容。具体流程图如图5-6所示。

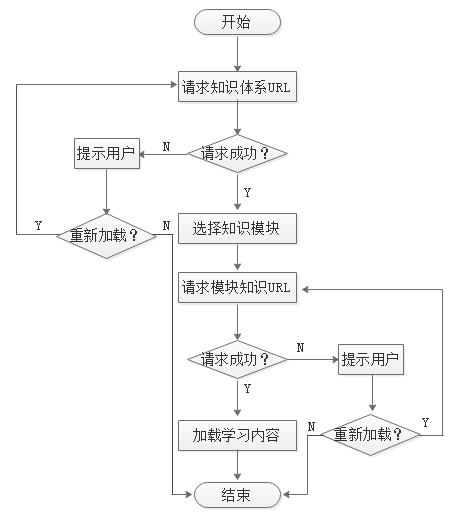


图5-6知识体系学习流程图

在知识体系功能模块设计中，Android客户端布局界面使用RecyclerView，列表中的每一项都是知识体系中的某一块知识。用户选择知识体系模块自动加载数据的部分关键代码如下所示：

public void getKnowledgeSystemData(boolean isShowViewStatus) {  
 addSubscribe(dataManager.getKnowledgeData()  
 .compose(RxJavaUtil.SchedulerTransformer())  
 .filter(listBaseResponse -> mView != null)  
 .subscribeWith(new

BaseObserver<List<KnowledgeSystemData>>(mView,

MobileLearnApp.getContext().getString(

R.string.failed\_to\_get\_knowledge),  
 isShowViewStatus) {  
 //知识体系数据请求成功后回调函数  
 @Override  
 public void success(List<KnowledgeSystemData>

knowledgeSystemData) {

mView.showKnowledgeSystemData(knowledgeSystemData);  
 }  
 }));  
 }

知识体系功能模块实现界面如图5-7。



图5-7 知识体系模块界面

### 5.2.4 公众号功能模块

公众号功能目的是让移动开发学习者对于技术类公众号的原创学习资源方便查看，将知名Android开发者、知名公司团队公众号集合在一起。公众号功能既能满足原创作者信息分享需求，又能满足移动开发学习者学习知识技能的需求。

当学习者如果遇到自己喜欢或者感兴趣的技术分享时，学习者可以先收藏起来，方便以后查找和学习。当学习者收藏文章时，首先判断用户是否处于登录状态，如果处于登录状态，需要根据文章的ID判断此文章是可收藏状态或是可取消状态，如果未处于登录状态，那么请先登录，具体流程图如图5-8所示。

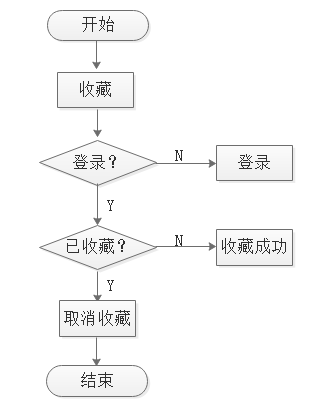


图5-8公众号文章收藏流程图

Android客户端首先通过请求公众号列表URL获取公众号列表数据，当请求的URL存在列表信息，PublicNumFragment类会调用相关的控件显示列表信息。当列表数据加载完毕后，系统自动加载列表中默认的个人公众号历史数据，用户可对感兴趣的内容收藏。用户点击收藏公众号文章部分关键代码如下所示：

private void childOnClick(View view, int position) {  
 if (presenter.getLoginStatus()) {  
 if (mAdapter.getData().get(position).isCollect()) {  
 presenter.cancelAddCollectArticle(position,  
 mAdapter.getData().get(position).getId());  
 } else {  
 presenter.addCollectArticle(position,

mAdapter.getData().get(position).getId());  
 }  
 } else {  
 Intent intent = new Intent(\_mActivity, LoginActivity.class);  
 startActivity(intent);  
 }  
}

公众号功能模块实现界面如图5-9。



图5-9公众号模块界面

### 5.2.5 项目功能模块

项目功能是Android开发学习者将开发知识综合运用并开发出一个完整的项目或某一功能Demo。学习者经过大量的理论知识学习，懂得了大部分概念，但如果不经过实践锻炼，学过的知识也会容易忘记。通过项目的学习，不仅有助于学习者对学过知识的理解，也利于学习者实际动手能力。

当学习者选择自己感兴趣的项目进行自学时，经常会遇到晦涩难懂的问题，学习者可通过分享的功能，将此项目分享给他人寻求帮助，能够让学习者解决疑难问题。学习者分享学习内容时，由于Android系统的权限管理机制从Android 6.0之后发生了较大的改变，有一些比较危险的权限要求必须申请动态权限，所以此系统也要进行权限申请。具体流程图如图5-10所示。

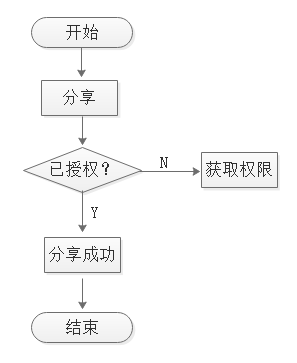


图5-10项目模块分享功能流程图

进入项目功能模块后，首先加载项目分类数据，当项目分类数据加载完毕后，客户端再次加载项目列表数据。此时，用户可以进入项目详情，在ToolBar的菜单里可以选择分享或用其他浏览器打开操作。用户分享部分关键代码如下所示：

public void checkSharePermission(RxPermissions rxPermissions) {  
 //检查是否授权  
 addSubscribe(rxPermissions  
 .request(Manifest.permission.WRITE\_EXTERNAL\_STORAGE)  
 .subscribe(new Consumer<Boolean>() {  
 @Override  
 public void accept(Boolean aBoolean) throws Exception

{  
 if (aBoolean) {  
 mView.shareArticle();  
 } else {  
 mView.shareFailed();  
 }  
 }  
 }));  
 }

项目功能模块实现界面如图5-11。



图5-11项目模块界面

### 5.2.6 个人中心功能模块

个人中心功能是学习者对个人信息的管理，学习者可以查看自己收藏过的文章，查看的前提条件必须要登录个人帐号。具体流程图如图5-12所示。

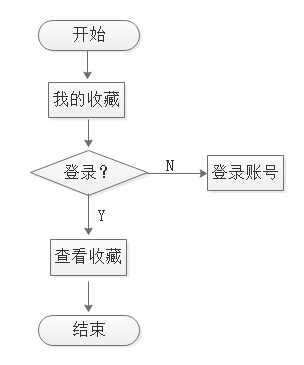


图5-12 查看我的收藏流程图

在实现中，系统通过调用Android中setVisible方法实现控件是否显示功能，当用户登录成功处于在线状态时，客户端才可以实现帐号注销操作，否则“注销”按钮将不会显示。按钮处理的部分关键代码如下所示：

public void handleLoginSuccess() {  
 tv\_welcome.setVisibility(View.VISIBLE);  
 tv\_login\_account.setText(mPresenter.getLoginAccount());  
 //隐藏登录按钮

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(false);  
 //显示注销按钮

navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(true);  
}

个人中心功能模块实现界面如图5-13。



图5-13个人中心模块界面

# 第六章 移动开发学习系统的测试

## 6.1 功能测试

移动开发学习系统采用黑盒测试的方法进行测试。黑盒测试别名为功能测试，主要是根据程序的设计规格说明书，在不考虑程序内部的逻辑结构和内部特性的情况下，测试人员检查程序功能是否符合规范。黑盒测试，在程序设计过程中被看作是一个打不开的盒子。同时，测试人员也看不到盒子中的程序代码。此次重点选择了在线学习功能、公众号文章收藏功能、用户注册功能等方面进行测试。

（1）在线学习功能测试

用户打开移动终端的移动开发学习系统APP,系统会自动进入在线学习平台的主页面，并进行相应测试。本文设计的在线学习功能测试用例，如表6-1所示。



表6-1在线学习功能测试用例

（2）公众号模块收藏功能测试

用户登录移动开发进入公众号功能模块，选择一篇学习资源查看详细内容，查看学习内容的同时可选择标题栏上的菜单图标进行分享操作。本文设计的公众号模块收藏功能测试用例，如表6-2所示。



表6-2公众号收藏功能测试用例

（3）用户登录注册功能

用户在未登录的状态下进入移动开发学习系统，此时进入个人中心功能模块通过选择注册按钮进行用户注册，注册成功后即可登录，登录验证通过显示个人账号信息。本文设计的用户登录注册功能测试用例，如表6-3所示。



表6-3登录、注册测试用例

## 6.2 性能测试

移动开发学习系统的性能测试方面，主要针对程序的健壮性和稳定性进行了测试。应用程序开发完成后，客户端的性能将由Android SDK自带的测试工具Monkey测试。Monkey是在Android操作系统中一个用Java语言编写的工具，在Android文件系统中的存放路径是：/system/framework/monkey.jar，是一款运行Android程序在一定压力下的自动化测试工具，主要测试APP是否会出现崩溃现象。

首先将adb命令配置到系统环境变量，连接手机，开启开发者选项模式，在命令行输入命令：adb shell monkey -p com.sunshinexu.mobilelearn 1000。该命令实现对移动开发学习系统客户端自动进行1000次用户点击、触摸屏输入、手势输入等伪随机事件。

经过测试，移动开发学习系统应用程序没有出现任何崩溃现象，说明移动客户端具有良好的稳定性和不错的性能。

# 第七章 总结

本文通过对在移动互联网环境下移动开发学习者的学习过程与学习方法进行研究与分析，为移动开发学习者提供了一个功能比较丰富的移动开发学习系统。

此系统充分发挥了移动互联网络的即时性、便捷性，随时为学习者提供最新，最全面的学习内容。系统设计有助于提高移动开发学习者的理论知识和增加实际项目经验，这也是系统相比于其他学习平台优势的一点。学习者可以通过知识导航在线学习，知识体系模块化学习，提供完整项目动手实践学习。还可以对自己兴趣的学习内容进行收藏，也可以对想学的知识进行搜索等。在次系统的设计实现部分主要移动开发学习为背景，具体主要工作如下。

（1）详细介绍了移动开发学习系统的研究背景及意义，分析了国内移动学习的研究现状，简述了基于Android操作系统的相关技术。

（2）对移动开发学习系统进行需求分析，分析其主要的功能性需求，确定系统需要实现的功能，分析其非功能性需求，目的达到应用程序的性能要求。

（3）对系统总体设计，确定移动开发学习系统架构，设计开发程序的架构。

（4）基于Android的移动学习系统详细的需求分析与总体设计，对系统进行了详细的设计，完成了移动学习系统的代码实现和系统相关测试。

本文中基于Android的移动开发学习系统的设计与实现，总体上完成了设计此系统的目标，为移动开发学习者提供了一个良好的学习平台，弥补了通过传统学习方式学习的缺陷和不足。

# 附录：源程序清单

/\*\*  
 \* 移动开发学习系统启动页面  
 \*/

public class SplashActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

//淡入淡出效果

overridePendingTransition(android.R.anim.*fade\_in*, android.R.anim.*fade\_out*);

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_splash*);

RelativeLayout rl\_splash = findViewById(R.id.*rl\_splash*);

//透明动画

AlphaAnimation alphaAnimation = new AlphaAnimation(0.3f, 1.0f);

alphaAnimation.setDuration(1000);

alphaAnimation.setAnimationListener(new

Animation.AnimationListener() {

@Override

public void onAnimationStart(Animation animation) {

}

@Override

public void onAnimationEnd(Animation animation) {

startActivity(new Intent(SplashActivity.this,

MainActivity.class));

finish();

overridePendingTransition(android.R.anim.*fade\_in*,android.R.anim.*fade\_out*);

}

@Override

public void onAnimationRepeat(Animation animation) {

}

});

rl\_splash.startAnimation(alphaAnimation);

}

}

/\*\*

\*移动开发学习系统布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/rl\_splash"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:id="@+id/tv\_splash"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="30dp"

android:textSize="18sp"

android:textColor="@color/black"

android:text=" @string/welcome\_word

"/>

</RelativeLayout>

public abstract class AbstractSimpleActivity extends SupportActivity {  
  
 private Unbinder unbinder;  
 private static final String *TAG* = "AbstractSimpleActivity";  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(getLayoutId());  
 //沉浸式状态栏  
 ImmersionBar.*with*(this)  
 .statusBarView(findViewById(R.id.*status\_bar*))  
 .statusBarDarkFont(true)  
 .keyboardEnable(true)  
 .init();  
 unbinder = ButterKnife.*bind*(this);  
 Log.*d*(*TAG*, "onCreate: ");  
 onViewCreate();  
 initToolbar();  
 initView();  
 initEventAndData();  
 }  
  
 protected abstract int getLayoutId();  
 protected abstract void onViewCreate();  
 protected abstract void initToolbar();  
 protected abstract void initView();  
 protected abstract void initEventAndData();  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 ImmersionBar.*with*(this).destroy();  
 if (unbinder != null && unbinder != Unbinder.*EMPTY*) {  
 unbinder.unbind();  
 unbinder = null;  
 }  
 }

}

/\*\*  
 \* 每个Activity都需要继承此类  
 \* @param<T>  
 \*/

public abstract class BaseActivity<T extends BasePresenter> extends AbstractSimpleActivity  
 implements HasSupportFragmentInjector, BaseView {  
 private static final String *TAG* = "BaseActivity";  
  
 @Inject  
 protected T mPresenter;  
 @Inject  
 DispatchingAndroidInjector<Fragment> dispatchingAndroidInjector;  
  
 @Override  
 protected void onCreate(@Nullable Bundle savedInstanceState) {  
 AndroidInjection.*inject*(this); //实现 Activity 的注入  
 super.onCreate(savedInstanceState);  
 Log.*d*(*TAG*, "onCreate: ");  
 }  
 /\*\*  
 \* 父类onCreate调用onViewCreate  
 \* 先执行父类onCreate，在执行子类实现的 onViewCreate  
 \* 最后执行子类的 onCreate  
 \*/@Override  
 protected void onViewCreate() {  
 Log.*d*(*TAG*, "onViewCreate: onViewCreate");  
 if (mPresenter != null) {  
 mPresenter.attachView(this);  
 }  
 }  
  
 @Override  
 public AndroidInjector<Fragment> supportFragmentInjector() {  
 return dispatchingAndroidInjector;  
 }  
  
 @Override  
 public void showErrorMessage(String message) {  
  
 }  
  
 @Override  
 public void showLoading() {  
  
 }  
  
 @Override  
 public void hideLoading() {  
  
 }  
  
 @Override  
 public void showError() {  
  
 }  
  
 @Override  
 public void showNoNetwork() {  
  
 }  
  
 @Override  
 public void showEmpty() {  
  
 }  
  
 @Override  
 public void showContent() {  
  
 }  
  
 @Override  
 public void handleLoginSuccess() {  
  
 }  
  
 @Override  
 public void handleLogoutSuccess() {  
  
 }  
}

/\*\*  
 \* View 基类接口  
 \*/public interface BaseView {  
 void showErrorMessage(String message);  
 void showLoading();  
 void hideLoading();  
 void showError();  
  
 void showNoNetwork();  
 void showEmpty();  
 void showContent();  
 void handleLoginSuccess();  
  
 void handleLogoutSuccess();  
}

/\*\*  
 \* Presenter 基类接口  
 \* @param <T>  
 \*/public interface BasePresenter<T extends BaseView> {  
  
 void attachView(T view);  
 void detachView();  
 void reload();  
 void registerEventBus();  
 void unregisterEventBus();  
  
 void setLoginStatus(boolean loginStatus);  
 boolean getLoginStatus();  
 String getLoginAccount();  
 void setLoginAccount(String account);  
  
}

/\*\*  
 \* 主页面  
 \*/public class MainActivity extends BaseActivity<MainPresenter>

implements MainContract.View {  
  
 private static final String *TAG* = "MainActivity";  
 @BindView(R.id.*drawer\_layout*)  
 DrawerLayout drawer\_layout;  
 @BindView(R.id.*toolbar*)  
 Toolbar toolbar;  
 @BindView(R.id.*toolbar\_title*)  
 TextView toolbar\_title;  
 @BindView(R.id.*navigation\_view*)  
 NavigationView navigation\_view;  
 @BindView(R.id.*bottom\_navigation\_view*)  
 BottomNavigationView bottom\_navigation\_view;  
  
 private int currentFgIndex = 0;  
 private int lastFgIndex = -1;  
  
 private HomePageFragment homePageFragment;  
 private NavigationFragment navigationFragment;  
 private KnowledgeFragment knowledgeFragment;  
 private PublicNumFragment publicNumFragment;  
 private ProjectFragment projectFragment;  
 private TextView tv\_welcome;  
 private TextView tv\_login\_account;  
 private long clickTime;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 Log.*d*(*TAG*, "onCreate: ");  
 }  
  
 protected int getLayoutId() {  
 return R.layout.*activity\_main*;  
 }  
  
 /\*\*  
 \* 初始化 Toolbar  
 \*/@Override  
 protected void initToolbar() {  
 setSupportActionBar(toolbar);  
 ActionBar actionBar = getSupportActionBar();  
 if (actionBar != null) {  
 actionBar.setDisplayShowTitleEnabled(false);  
 toolbar\_title.setText(R.string.*home\_page*);  
 }

getWindow().setNavigationBarColor(getResources().getColor(R.colo r.*navigation\_bar*));

}  
  
 @Override  
 protected void initView() {  
 initDrawerLayout();  
 showFragment(currentFgIndex);  
 initNavigationView();  
 initBottomNavigationView();  
  
 }  
  
 /\*\*  
 \* 初始化底部导航栏  
 \*/private void initBottomNavigationView() {

bottom\_navigation\_view.setOnNavigationItemSelectedListener(new

BottomNavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem

menuItem) {  
 switch (menuItem.getItemId()) {  
 case R.id.*tab\_home\_pager*:  
 showFragment(Constants.*FRAGMENT\_HOME\_PAGE*);  
 break;  
 case R.id.*tab\_knowledge*:  
 showFragment(Constants.*FRAGMENT\_KNOWLEDGE*);  
 break;  
 case R.id.*tab\_navigation*:  
 showFragment(Constants.*FRAGMENT\_NAIVIGATION*);  
 break;  
 case R.id.*tab\_project*:  
 showFragment(Constants.*FRAGMENT\_PRPJECT*);  
 break;  
 case R.id.*tab\_public\_num*:  
 showFragment(Constants.*FRAGMENT\_PUBLIC\_NUM*);  
 break;  
 }  
 return true;  
 }  
 });  
 }  
  
 private void initNavigationView() {  
 navigation\_view.setNavigationItemSelectedListener(new

NavigationView.OnNavigationItemSelectedListener() {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem

menuItem) {  
 switch (menuItem.getItemId()) {  
 case R.id.*item\_login*:  
 startLoginActivity();  
 break;  
 case R.id.*item\_collect*:  
 if (mPresenter.getLoginStatus()) {   
 Intent intent = new

Intent(MainActivity.this, ItemActivity.class);  
 intent.putExtra("Type",

Constants.*TYPE\_COLLECT*);  
 startActivity(intent);  
 } else {  
 startLoginActivity(); //未登录跳转登录页  
 Toast.*makeText*(MainActivity.this,

getString(R.string.*please\_login*),  
 Toast.*LENGTH\_SHORT*).show();  
 }  
 break;  
 case R.id.*item\_about*:  
 Intent intentAbout = new

Intent(MainActivity.this,

ItemActivity.class);intentAbout.putExtra("Typ e",Constants.*TYPE\_ABOUT*);  
 startActivity(intentAbout);  
 break;  
 case R.id.*item\_logout*:  
 mPresenter.logout();  
 break;  
 }  
 return true;  
 }  
 });  
 tv\_welcome = navigation\_view.getHeaderView(0)

.findViewById(R.id.*tv\_welcome*);  
tv\_login\_account = navigation\_view.getHeaderView(0)

.findViewById(R.id.*tv\_login\_account*);

if (mPresenter.getLoginStatus()) {  
 tv\_welcome.setVisibility(View.*VISIBLE*);  
 tv\_login\_account.setText(mPresenter.getLoginAccount());

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(false); navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(true);  
} else {  
 tv\_welcome.setVisibility(View.*INVISIBLE*);  
 tv\_login\_account.setText("");

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(true); navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(false);

}

}  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.*toolbar\_menu*, menu);  
 return super.onCreateOptionsMenu(menu);  
 }  
  
 /\*\*  
 \* Toolbar 两个按钮  
 \* @param item  
 \* @return\*/@Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 Intent intent;  
 switch (item.getItemId()) {  
 case R.id.*action\_usage*:  
 intent=new

Intent(MainActivity.this, ItemActivity.class);  
 intent.putExtra("Type", Constants.*TYPE\_WEBSITE*);  
 startActivity(intent);  
 break;  
 case R.id.*action\_search*:

intent=new.

Intent(MainActivity.this,SearchActivity.class);

startActivity(intent);  
 break;  
 default:  
 break;  
 }  
 return true;  
 }  
  
 /\*\*  
 \* 展示对应 Fragment  
 \* @paramindex  
 \*/private void showFragment(int index) {  
 currentFgIndex = index;  
 FragmentTransaction transaction = getSupportFragmentManager()

.beginTransaction();

hideFragment(transaction);  
 lastFgIndex = index;  
 switch (index) {  
 case Constants.*FRAGMENT\_HOME\_PAGE*:  
 toolbar\_title.setText(R.string.*home\_page*);  
 if (homePageFragment == null) {  
 homePageFragment = HomePageFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.homePageFragment);  
 }  
 transaction.show(this.homePageFragment);  
 break;  
 case Constants.*FRAGMENT\_PUBLIC\_NUM*:  
 toolbar\_title.setText(R.string.*public\_num*);  
 if (publicNumFragment == null) {  
 publicNumFragment =

PublicNumFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.publicNumFragment);  
 }  
 transaction.show(this.publicNumFragment);  
 break;  
 case Constants.*FRAGMENT\_KNOWLEDGE*:  
 toolbar\_title.setText(R.string.*knowledge*);  
 if (knowledgeFragment == null) {  
 knowledgeFragment =

KnowledgeFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.knowledgeFragment);  
 }  
 transaction.show(this.knowledgeFragment);  
 break;  
 case Constants.*FRAGMENT\_PRPJECT*:  
 toolbar\_title.setText(R.string.*project*);  
 if (projectFragment == null) {  
 projectFragment = ProjectFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.projectFragment);  
 }  
 transaction.show(this.projectFragment);  
 break;  
 case Constants.*FRAGMENT\_NAIVIGATION*:  
 toolbar\_title.setText(R.string.*navigation*);  
 if (navigationFragment == null) {  
 navigationFragment =

NavigationFragment.*newInstance*();  
 transaction.add(R.id.*fragment\_root*,

this.navigationFragment);  
 }  
 transaction.show(this.navigationFragment);  
 break;  
 default:  
 break;  
 }  
 transaction.commit();  
 }  
  
 private void hideFragment(FragmentTransaction transaction) {  
 switch (lastFgIndex) {  
 case Constants.*FRAGMENT\_HOME\_PAGE*:  
 if (homePageFragment != null) {  
 transaction.hide(homePageFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_NAIVIGATION*:  
 if (navigationFragment != null) {  
 transaction.hide(navigationFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_KNOWLEDGE*:  
 if (knowledgeFragment != null) {  
 transaction.hide(knowledgeFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_PUBLIC\_NUM*:  
 if (publicNumFragment != null) {  
 transaction.hide(publicNumFragment);  
 }  
 break;  
 case Constants.*FRAGMENT\_PRPJECT*:  
 if (projectFragment != null) {  
 transaction.hide(projectFragment);  
 }  
 break;  
 default:  
 break;  
 }  
 }  
  
 private void initDrawerLayout() {  
 ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this,

drawer\_layout, toolbar,

R.string.*open\_drawer*, R.string.*close\_drawer*);  
 toggle.syncState();  
 drawer\_layout.addDrawerListener(toggle);  
 }  
  
 private void startLoginActivity() {  
 Intent intent = new Intent(this, LoginActivity.class);  
 startActivity(intent);  
 }  
  
  
 @Override  
 protected void initEventAndData() {  
  
 }  
  
  
 @Override  
 public void showErrorMessage(String message) {  
  
 }  
  
 @Override  
 public void showLoading() {  
  
 }  
  
 @Override  
 public void hideLoading() {  
  
 }  
  
 @Override  
 public void showError() {  
  
 }  
  
 @Override  
 public void showNoNetwork() {  
  
 }  
  
 @Override  
 public void showEmpty() {  
  
 }  
  
 @Override  
 public void showContent() {  
  
 }  
  
 @Override  
 public void handleLoginSuccess() {  
 tv\_welcome.setVisibility(View.*VISIBLE*);  
 tv\_login\_account.setText(mPresenter.getLoginAccount());  
 //隐藏登录按钮

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(false);  
 //显示注销按钮

navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(true);  
 }  
  
 @Override  
 public void handleLogoutSuccess() {  
 tv\_welcome.setVisibility(View.*INVISIBLE*);  
 tv\_login\_account.setText("");

navigation\_view.getMenu().findItem(R.id.*item\_login*)

.setVisible(true);

navigation\_view.getMenu().findItem(R.id.*item\_logout*)

.setVisible(false);  
Toast.*makeText*(this,getString(R.string.*logout\_success*)

,Toast.*LENGTH\_SHORT*).show();

}  
 */*\*\*  
 \* 处理回退事件  
 \*/@Override  
 public void onBackPressedSupport() {  
 if (getSupportFragmentManager().getBackStackEntryCount() > 1)

{  
 pop();  
 } else {  
 long currentTime = System.*currentTimeMillis*();  
 if((currentTime - clickTime) >

Constants.*DOUBLE\_INTERVAL\_TIME*)

{  
 Toast.*makeText*(MainActivity.this,

getString(R.string.*double\_click\_exit\_toast*)

,Toast.*LENGTH\_SHORT*).show();  
 clickTime = System.*currentTimeMillis*();  
 } else {  
 finish();  
 }  
 }  
 }  
}

/\*\*

\* Toolbar 布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<!--支持响应滚动手势的 bar-->

<android.support.design.widget.AppBarLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:elevation='0dp'>

<View

android:id="@+id/status\_bar"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:background="@color/Grey250"/>

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

style="@style/ToolBarStyle"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="@color/Grey250"

android:gravity="center\_vertical"

app:contentInsetStart="0dp"

app:contentInsetEnd="0dp"

app:contentInsetStartWithNavigation="0dp"

app:layout\_scrollFlags="scroll|enterAlways|snap">

<ImageView

android:id="@+id/toolbar\_back"

android:layout\_width="24dp"

android:layout\_height="24dp"

android:layout\_marginLeft="15dp"

android:layout\_marginRight="10dp"

android:src="@drawable/ic\_left\_black"

android:visibility="gone"/>

<TextView

android:id="@+id/toolbar\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="start"

android:ellipsize="marquee"

android:singleLine="true"

android:textSize="16sp"

android:textStyle="bold"

android:textColor="@color/black"

/>

</android.support.v7.widget.Toolbar>

</android.support.design.widget.AppBarLayout>

/\*\*

\* 主页面布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/drawer\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<android.support.design.widget.CoordinatorLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<include layout="@layout/toolbar" />

<FrameLayout

android:id="@+id/fragment\_root"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

</FrameLayout>

<android.support.design.widget.BottomNavigationView

android:id="@+id/bottom\_navigation\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom"

android:background="@color/white"

app:elevation="10dp"

app:itemIconTint="@drawable/navigation\_bottom\_color"

app:itemTextColor="@drawable/navigation\_bottom\_color"

app:labelVisibilityMode="labeled"

app:menu="@menu/bottom\_navigation">

</android.support.design.widget.BottomNavigationView>

</android.support.design.widget.CoordinatorLayout>

<android.support.design.widget.NavigationView

android:id="@+id/navigation\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

android:background="@color/main\_white"

app:headerLayout="@layout/navigation\_header"

app:insetForeground="@android:color/transparent"

app:itemIconTint="@color/navigation\_item\_ic\_color"

app:itemTextColor="@color/black"

app:menu="@menu/navigation\_view">

</android.support.design.widget.NavigationView>

</android.support.v4.widget.DrawerLayout>

public abstract class AbstractSimpleFragment extends SupportFragment {

private Unbinder unbinder;

private static final String TAG = "AbstractSimpleFragment";

@Override

public void onAttach(Activity activity) {

Log.d(TAG, "onAttach: ");

super.onAttach(activity);

}

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater,

@Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

View view = inflater.inflate(getLayoutId(),container,false);

Log.d(TAG, "onCreateView1: ");

unbinder = ButterKnife.bind(this,view);

initView();

Log.d(TAG, "onCreateView: ");

return view;

}

@Override

public void onLazyInitView(@Nullable Bundle savedInstanceState) {

super.onLazyInitView(savedInstanceState);

initData();

}

@Override

public void onDestroyView() {

super.onDestroyView();

if (unbinder != null && unbinder != Unbinder.EMPTY ) {

unbinder.unbind();

unbinder = null;

}

}

/\*\*

\*Fragment 销毁时进行内存泄漏检测

\*/

@Override

public void onDestroy() {

super.onDestroy();

RefWatcher refWatcher =

MobileLearnApp.getRefWatcher(\_mActivity);

refWatcher.watch(this);

}

protected abstract int getLayoutId();

protected abstract void initView();

protected abstract void initData();

}

/\*\*

\* 每个Fragment都需要继承此类

\* @param <T>

\*/

public abstract class BaseFragment<T extends BasePresenter> extends AbstractSimpleFragment implements BaseView {

private static final String TAG = "BaseFragment";

@Inject

protected T presenter;

private MultipleStatusView custom\_multiple\_view;

private LoadingDialog dialog;

@Override

public void onAttach(Activity activity) {

Log.d(TAG, "onAttach: ");

AndroidSupportInjection.inject(this);

//必须在 onAttach 中完成

super.onAttach(activity);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle. savedInstanceState) {

Log.d(TAG, "onViewCreated: ");

super.onViewCreated(view, savedInstanceState);

custom\_multiple\_view =. view.findViewById(R.id.custom\_multiple\_view);

if (presenter != null) {

presenter.attachView(this);

}

if (custom\_multiple\_view != null) {

custom\_multiple\_view.setOnRetryClickListener(v ->. presenter.reload());

}

}

@Override

public void onDetach() {

super.onDetach();

if (presenter != null) {

presenter = null;

}

}

@Override

public void onDestroyView() {

super.onDestroyView();

if (presenter != null) {

presenter.detachView();

}

}

@Override

public void showErrorMessage(String message) {

Toast.makeText(\_mActivity, message,

Toast.LENGTH\_SHORT).show();

}

/\*\*

\* 弹出加载框

\*/

@Override

public void showLoading() {

System.out.println("showdialog");

if (custom\_multiple\_view == null) {

return;

} else {

System.out.println("showdialog");

if (dialog == null) {

dialog=newLoadingDialog(\_mActivity, getString(R.string.loading\_data),

R.mipmap.ic\_dialog\_loading);

}

dialog.show();

}

}

/\*\*

\* 隐藏加载框

\*/

@Override

public void hideLoading() {

if (dialog != null) {

System.out.println("hidedialog");

dialog.dismiss();

dialog = null;

}

}

@Override

public void showError() {

if (custom\_multiple\_view == null) return;

custom\_multiple\_view.showError();

}

@Override

public void showNoNetwork() {

if (custom\_multiple\_view == null) return;

custom\_multiple\_view.showNoNetwork();

}

@Override

public void showEmpty() {

if (custom\_multiple\_view == null) return;

custom\_multiple\_view.showEmpty();

}

@Override

public void showContent() {

if (custom\_multiple\_view == null) {

return;

} else {

custom\_multiple\_view.showContent();

}

}

@Override

public void handleLoginSuccess() {

}

@Override

public void handleLogoutSuccess() {

}

}

public interface CollectContract {

interface View extends BaseView {

void showCollectSuccess(int position);

void showCancelCollect(int position);

}

interface Presenter<V extends View> extends BasePresenter<V> {

void addCollectArticle(int position, int id);

void cancelAddCollectArticle(int position, int id);

}

}

/\*\*

\* 关于收藏相关

\*/

public class CollectPresenter<V extends CollectContract.View> extends Presenter<V>implements CollectContract.Presenter<V> {

@Override

public void addCollectArticle(int position, int id) {

addSubscribe(dataManager.addArticleCollect(id)

.compose(RxJavaUtil.SchedulerTransformer())//调度器

.filter(articleListData -> mView != null) //添加过滤

.subscribeWith(new. BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString

(R.string.collect\_failed),false) {

@Override

Public

void success(ArticleListData articleListData) {

mView.showCollectSuccess(position);

}

}));

}

@Override

public void cancelAddCollectArticle(int position, int id) {

addSubscribe(dataManager.cancelAddArticleCollect(id)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new. BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString

(R.string.failed\_to\_cancel\_collect)) {

@Override

public

void success(ArticleListData articleListData) {

mView.showCancelCollect(position);

}

}));

}

}

/\*\*

\* 在线学习模块相关调用

\*/

Public class HomepagePresenter extends

CollectPresenter<HomepageContract

.View implements HomepageContract.Presenter {

private int currentPage;

private boolean isRefresh;

@Inject

HomepagePresenter() {

}

@Override

public void getArticleData(boolean isShowStatusView) {

addSubscribe(dataManager.getArticleList(currentPage)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new

BaseObserver<ArticleListData>(mView, MobileLearnApp .getContext().getString(R.string.failed\_to\_get\_article), isShowStatusView) {

@Override

public void success(ArticleListData

articleListData) {

mView.showList(articleListData, isRefresh);

}

}));

}

/\*\*

\* 获取首页的 Banner 数据

\* @param isShowStatusView

\*/

@Override

public void getBannerData(boolean isShowStatusView) {

addSubscribe(dataManager.getBannerData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleBannerData -> mView != null)

.subscribeWith(new

BaseObserver<List<BannerData>>(mView,

MobileLearnApp.getContext().getString(R.string.failed \_to\_get\_banner),isShowStatusView) {

@Override

public void success(List<BannerData> bannerData) {

mView.showBannerData(bannerData);

}

}));

}

/\*\*

\* zip() 一个函数组合多个 Observable 发射的数据集合

\* 可将多个请求结合

\* lambda 接收两个参数

\* subscribeWith 一般使用 Rx 请求接口的这种情况会用

\* @param isShowStatusView

\*/

@Override

public void getHomepageData(boolean isShowStatusView) {

getBannerData(isShowStatusView);

addSubscribe(Observable.zip(dataManager.getUpArticles(),

dataManager.getArticleList(0),

(upArticlesResponse, articleListResponse) -> {

articleListResponse.getData().getDatas().addAll(0,

upArticlesResponse.getData());

return articleListResponse;

})

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new

BaseObserver<ArticleListData>(mView, MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_article), isShowStatusView) {

@Override

public void success(ArticleListData

articleListData) {

mView.showList(articleListData, isRefresh);

}

}));

}

/\*\*

\* 下拉刷新

\* @param isShowStatusView

\*/

@Override

public void refreshLayout(boolean isShowStatusView) {

isRefresh = true;

currentPage = 0;

getHomepageData(isShowStatusView);

}

/\*\*

\* 下拉加载更多

\*/

@Override

public void loadMore() {

isRefresh = false;

currentPage++;

getArticleData(false);

}

/\*\*

\* 注销成功后刷新数据

\* @param logoutEvent

\*/

@Subscriber()

public void logoutSuccessEvent(LogoutEvent logoutEvent) {

getHomepageData(false);

}

/\*\*

\* 登录成功后刷新数据

\* @param loginEvent

\*/

@Subscriber()

public void loginSuccessEvent(LoginEvent loginEvent) {

getHomepageData(false);

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

}

/\*\*

\* 在线学习模块 Fragment

\*/

public class HomePageFragment extends BaseFragment<HomepagePresenter> implements HomepageContract.View {

private static final String TAG = "HomePageFragment";

@BindView(R.id.hp\_recycler\_view)

RecyclerView hpRecyclerView;

@BindView(R.id.hp\_refresh\_layout)

SmartRefreshLayout hpSmartRefreshLayout;

private Banner banner;

private ArticleListAdapter adapter;

public static HomePageFragment newInstance() {

HomePageFragment homePageFragment = new HomePageFragment();

return homePageFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_homepage;

}

@Override

protected void initView() {

List<ArticleItemData> articleList = new ArrayList<>();

Adapter = new ArticleListAdapter

(R.layout.item\_article.articleList)

adapter.setOnItemClickListener(new. BaseQuickAdapter.OnItemClickListener() {

@Override

public void onItemClick(BaseQuickAdapter adapter, View

view, int position) {

openArticleDetail(view, position);

}

});

adapter.setOnItemChildClickListener(new

BaseQuickAdapter.OnItemChildClickListener() {

@Override

public void onItemChildClick(BaseQuickAdapter adapter,

View view, int position) {

childOnClick(view, position);

}

});

hpRecyclerView.setLayoutManager(new. LinearLayoutManager(\_mActivity));

DividerItemDecorationdivider = newDividerItemDecoration

(\_mActivity,DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,R.drawable.shape\_hp\_rv\_line));

hpRecyclerView.addItemDecoration(divider);

hpRecyclerView.setHasFixedSize(true);

//添加头布局 Banner 支持轮播效果

LinearLayout mGroup = (LinearLayout) getLayoutInflater()

.inflate(R.layout.head\_banner, null);

banner = mGroup.findViewById(R.id.head\_banner);

mGroup.removeView(banner);

adapter.setHeaderView(banner);

hpRecyclerView.setAdapter(adapter);

}

private void childOnClick(View view, int position) {

switch (view.getId()) {

case R.id.iv\_article\_like:

collectArticle(position);

break;

default:

break;

}

}

/\*\*

\* 收藏文章

\*

\* @param position 传入点击的位置

\*/

private void collectArticle(int position) {

if (presenter.getLoginStatus()) {

if (adapter.getData().get(position).isCollect()) {

presenter.cancelAddCollectArticle(position,adapter.getData().get(position).getId());

} else {

presenter.addCollectArticle(position,adapter.getDat a().get(position).getId());

}

} else {

Intent intent = new Intent(\_mActivity,

LoginActivity.class);

startActivity(intent);

}

}

/\*\*

\* 打开文章详情

\* @param view

\* @param position

\*/

private void openArticleDetail(View view, int position) {

Intent intent = new Intent

(\_mActivity, ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,adapter.getData().get(position).getLink()); intent.putExtra(Constants.ARTICLE\_TITLE,adapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,adapter.getData().get(position).getId()); intent.putExtra(Constants.IS\_COLLECTED,adapter.getData().get(position).isCollect());

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,position);

intent.putExtra(Constants.EVENT\_BUS\_TAG,"homepage");

startActivity(intent);

}

@Override

protected void initData() {

initRefreshLayout();

presenter.refreshLayout(true);

}

private void initRefreshLayout() {

//下拉刷新

hpSmartRefreshLayout.setOnRefreshListener(new. OnRefreshListener() {

@Override

public void onRefresh(@NonNull RefreshLayout refreshLayout)

{

presenter.refreshLayout(false);

refreshLayout.finishRefresh();

}

});

//上拉加载更多

hpSmartRefreshLayout.setOnLoadMoreListener(new. OnLoadMoreListener() {

@Override

public void onLoadMore(@NonNull RefreshLayout

refreshLayout) {

presenter.loadMore();

refreshLayout.finishLoadMore();

}

});

}

/\*\*

\* 展示文章列表

\* @param listData 文章列表 list

\* @param isRefresh 是否刷新

\*/

@Override

public void showList(ArticleListData listData, boolean isRefresh). {

if (isRefresh) {

adapter.replaceData(listData.getDatas());

} else {

adapter.addData(listData.getDatas());

}

}

/\*\*

\* 展示轮播图

\* @param bannerDataList 轮播图 list

\*/

@Override

public void showBannerData(List<BannerData> bannerDataList) {

ArrayList<String> imageList = new ArrayList<>();

ArrayList<String> titleList = new ArrayList<>();

ArrayList<String> urlList = new ArrayList<>();

ArrayList<Integer> articleIdList = new ArrayList<>();

for (BannerData bannerData : bannerDataList) {

imageList.add(bannerData.getImagePath());

titleList.add(bannerData.getTitle());

urlList.add(bannerData.getUrl());

articleIdList.add(bannerData.getId());

}

//显示圆形指示器和标题

banner.setBannerStyle(BannerConfig.CIRCLE\_INDICATOR\_TITLE\_INSIDE);

banner.setImageLoader(new GlideImageLoader());

banner.setImages(imageList);

banner.setBannerAnimation(Transformer.ZoomOutSlide);

banner.setBannerTitles(titleList);

banner.isAutoPlay(true);

banner.setDelayTime(2800);

banner.setIndicatorGravity(BannerConfig.CENTER);

banner.start();

banner.setOnBannerListener(new OnBannerListener() {

@Override

public void OnBannerClick(int position) {

Intent intent = new Intent

(\_mActivity, ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,urlList.get

(posiion));

intent.putExtra(Constants.ARTICLE\_TITLE,titleList.get

(position));

intent.putExtra(Constants.ARTICLE\_ID,articleIdList.get

(position));

intent.putExtra(Constants.IS\_COLLECTED,false);

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,false);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,-1);

Intent.putExtra(Constants.EVENT\_BUS\_TAG,

Constants.TAG\_DEFAULT);

startActivity(intent);

}

});

}

/\*\*

\* 收藏成功

\* @param position 收藏位置

\*/

@Override

public void showCollectSuccess(int position) {

adapter.getData().get(position).setCollect(true);

adapter.setData(position,adapter.getData().get(position));

Toast.makeText(\_mActivity,R.string.collect\_success,Toast.LENGTH\_SHORT).show();

}

/\*\*

\* 取消收藏

\* @param position 取消位置

\*/

@Override

public void showCancelCollect(int position) {

adapter.getData().get(position).setCollect(false);

adapter.setData(position,adapter.getData().get(position));

Toast.makeText(\_mActivity,R.string.cancel\_collect,Toast.LENGTH\_SHORT).show();

}

}

/\*\*

\* 在线学习模块布局文件

\*/

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/hp\_refresh\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="@color/colorAccent">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

</com.scwang.smartrefresh.header.MaterialHeader>

<android.support.v7.widget.RecyclerView

android:id="@+id/hp\_recycler\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

</com.scwang.smartrefresh.header.MaterialHeader>

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 学习文章详情页

\*/

public class ArticleDetailActivity extends BaseActivity

<ArticleDetailPresenter> implements ArticleDetailContract.View {

private String title;

private String articleLink;

private int articleId;

private boolean isCollected;

private boolean isShowCollectIcon;

private int articleItemPosition;

private String eventBusTag;

@BindView(R.id.toolbar)

Toolbar toolbar;

@BindView(R.id.toolbar\_title)

TextView toolbarTitle;

@BindView(R.id.toolbar\_back)

ImageView toolbar\_back;

@BindView(R.id.cl\_content)

CoordinatorLayout cl\_content;

private AgentWeb agentWeb;

private MenuItem collectItem;

@Override

protected int getLayoutId() {

return R.layout.activity\_article\_detail;

}

@Override

protected void initToolbar() {

getIntentData();

getWindow().setNavigationBarColor(getResources()

.getColor(R.color.white\_smoke));

toolbar.setNavigationIcon(R.drawable.ic\_arrow\_back\_black\_24dp);

setSupportActionBar(toolbar);

ActionBar actionBar = getSupportActionBar();

if (actionBar != null) {

//左上角图标左边加上返回图标

actionBar.setDisplayHomeAsUpEnabled(false);

toolbar\_back.setVisibility(View.VISIBLE);

actionBar.setDisplayShowTitleEnabled(false);

toolbarTitle.setText(Html.fromHtml(title));

toolbarTitle.setSelected(true);

}

toolbar.setNavigationOnClickListener(new. View.OnClickListener() {

@Override

public void onClick(View view) {

onBackPressedSupport();

}

});

toolbar\_back.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

finish();

}

});

}

private void getIntentData() {

Bundle bundle = getIntent().getExtras();

title = bundle.getString(Constants.ARTICLE\_TITLE);

articleLink = bundle.getString(Constants.ARTICLE\_LINK);

articleId = bundle.getInt(Constants.ARTICLE\_ID);

isCollected = bundle.getBoolean(Constants.IS\_COLLECTED);

isShowCollectIcon = bundle.getBoolean

(Constants.IS\_SHOW\_COLLECT\_ICON);

articleItemPosition = bundle.getInt

(Constants.ARTICLE\_ITEM\_POSITION, -1);

eventBusTag = bundle.getString(Constants.EVENT\_BUS\_TAG);

}

@Override

protected void initView() {

}

@Override

protected void initEventAndData() {

//集合了影响浏览器的事件到来时的回调方法

WebChromeClient webChromeClient = new WebChromeClient() {

@Override

public void onReceivedTitle(WebView view, String title) {

super.onReceivedTitle(view, title);

// toolbar.setTitle(Html.fromHtml(title));

}

};

CoordinatorLayout.LayoutParams layoutParams = new

CoordinatorLayout.LayoutParams(-1, -1);

layoutParams.setBehavior(new. AppBarLayout.ScrollingViewBehavior());

NestedScrollAgentWebView nestedScrollAgentWebView = new

NestedScrollAgentWebView(this);

//AgentWeb 是对 WebView 的封装

agentWeb = AgentWeb.with(this)

.setAgentWebParent(cl\_content, layoutParams)

.useDefaultIndicator()

.setWebView(nestedScrollAgentWebView)

.setWebChromeClient(webChromeClient)

.setMainFrameErrorView(R.layout.agentweb\_error\_page, -1)

.setOpenOtherPageWays(DefaultWebClient.OpenOtherPageW ays.ASK)

.createAgentWeb()

.ready()

.go(articleLink);

}

/\*\*

\* 创建 ToolBar 上的菜单

\*

\* @param menu 菜单样式

\* @return

\*/

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.article\_detail\_menu, menu);

collectItem = menu.findItem(R.id.item\_collect);

collectItem.setVisible(isShowCollectIcon);

collectItem.setIcon(isCollected ? R.drawable.ic\_collect :

R.drawable.ic\_collect\_not);

return super.onCreateOptionsMenu(menu);

}

/\*\*

\* 可以让打开的菜单同时显示图标和文字

\*

\* @param featureId

\* @param menu

\* @return

\*/

@Override

public boolean onMenuOpened(int featureId, Menu menu) {

if (menu != null) {

if("MenuBuilder".equalsIgnoreCase(menu.getClass().getSimpleName())) {

try {

@SuppressLint("PrivateApi")

Method method = menu.getClass()

.getDeclaredMethod("setOptionalIconsVisible",

Boolean.TYPE);

method.setAccessible(true);

method.invoke(menu, true);

} catch (Exception e) {

e.printStackTrace();

}

}

}

return super.onMenuOpened(featureId, menu);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.item\_collect:

dealCollect();

break;

case R.id.item\_share:

mPresenter.checkSharePermission(new RxPermissions(this));

break;

case R.id.item\_open\_browser:

Intent intent = new Intent(Intent.ACTION\_VIEW,

Uri.parse(articleLink));

startActivity(intent);

break;

default:

break;

}

return super.onOptionsItemSelected(item);

}

private void dealCollect() {

if (mPresenter.getLoginStatus()) {

if (isCollected) {

mPresenter.cancelAddCollectArticle(articleItemPosition,

articleId);

} else {

mPresenter.addCollectArticle(articleItemPosition,

articleId);

}

} else {

Intent intent = new Intent(this, LoginActivity.class);

startActivity(intent);

}

}

/\*\*

\* 分享学习资源

\*/

@Override

public void shareArticle() {

Intent intent = new Intent(Intent.ACTION\_SEND);

intent.putExtra(Intent.EXTRA\_TEXT, getString(R.string.share\_type\_url, "MobileLearn",

title, articleLink));

intent.setType("text/plain");

startActivity(intent);

}

@Override

public void shareFailed() {

Toast.makeText(this, R.string.failed\_to\_share\_failed,

Toast.LENGTH\_SHORT).show();

}

@Override

public void showCollectSuccess(int position) {

isCollected = true;

collectItem.setIcon(R.drawable.ic\_collect);

if (position < 0) {

Toast.makeText(this, R.string.collect\_success,

Toast.LENGTH\_SHORT).show();

} else {

EventBus.getDefault().post(new CollectEvent(false,

position), eventBusTag);

}

}

@Override

public void showCancelCollect(int position) {

isCollected = false;

collectItem.setIcon(R.drawable.ic\_collect\_not);

if (position < 0) {

Toast.makeText(this, R.string.collect\_failed,

Toast.LENGTH\_SHORT).show();

} else {

EventBus.getDefault().post(new CollectEvent(true,

position), eventBusTag);

}

}

@Override

public void onBackPressedSupport() {

if (!agentWeb.back()) {

super.onBackPressedSupport();

}

}

@Override

protected void onResume() {

agentWeb.getWebLifeCycle().onResume();

super.onResume();

}

@Override

protected void onPause() {

agentWeb.getWebLifeCycle().onPause();

super.onPause();

}

@Override

protected void onDestroy() {

agentWeb.getWebLifeCycle().onDestroy();

super.onDestroy();

}

}

/\*\*

\* 学习资源详情页布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/cl\_content"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

tools:context=".activity.main.ui.ArticleDetailActivity">

<include layout="@layout/toolbar"/>

</android.support.design.widget.CoordinatorLayout>

/\*\*

\* 学习导航相关调用

\*/

public class NavigationPresenter extends Presenter

<NavigationContract.View> implements NavigationContract.Presenter {

@Inject

public NavigationPresenter() {

}

@Override

public void getNavigationData() {

addSubscribe(dataManager.getNavigationData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(navigationListData -> mView != null)

.subscribeWith(new

BaseObserver<List<NavigationData>>(mView,

MobileLearnApp.getContext().getString

(R.string.failed\_to\_get\_navigation\_data), true) {

@Override

public void success(List<NavigationData>

navigationData) {

mView.showNavigationData(navigationData);

}

}));

}

@Override

public void reload() {

getNavigationData();

}

}

/\*\*

\* 学习导航页面

\*/

public class NavigationFragment extends BaseFragment

<NavigationPresenter> implements NavigationContract.View {

@BindView(R.id.rv\_navigation)

RecyclerView rv\_navigation;

@BindView(R.id.vtl\_navigation)

VerticalTabLayout vtl\_navigation;

private NavigationAdapter adapter;

private boolean scroll;

private int index;

private boolean isClickTab;

private LinearLayoutManager mLinearLayoutManager;

public static NavigationFragment newInstance() {

return new NavigationFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_navigation;

}

@Override

protected void initView() {

List<NavigationData> navigationDataList = new ArrayList<>();

adapter = new NavigationAdapter(R.layout.item\_navigation,

navigationDataList);

DividerItemDecoration divider = new DividerItemDecoration

(\_mActivity, DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable

(\_mActivity, R.drawable.shape\_hp\_rv\_line));

mLinearLayoutManager = new LinearLayoutManager(\_mActivity);

rv\_navigation.setLayoutManager(mLinearLayoutManager);

rv\_navigation.setHasFixedSize(true);

rv\_navigation.setAdapter(adapter);

}

@Override

protected void initData() {

presenter.getNavigationData();

}

/\*\*

\* 展示学习导航资源

\* @param navigationData 导航数据

\*/

@Override

public void showNavigationData(List<NavigationData>

navigationData) {

vtl\_navigation.setTabAdapter(new TabAdapter() {

@Override

public int getCount() {

return navigationData == null ? 0 :

navigationData.size();

}

@Override

public ITabView.TabBadge getBadge(int i) {

return null;

}

@Override

public ITabView.TabIcon getIcon(int i) {

return null;

}

@Override

public ITabView.TabTitle getTitle(int i) {

return new TabView.TabTitle.Builder()

.setContent(navigationData.get(i).getName())

.setTextColor(ContextCompat.getColor(\_mActivi ty, R.color.Grey1000),

ContextCompat.getColor(\_mActivity,

R.color.bottom\_bar))

.build();

}

@Override

public int getBackground(int i) {

return -1;

}

});

adapter.replaceData(navigationData);

rv\_navigation.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrollStateChanged(RecyclerView

recyclerView, int newState) {

super.onScrollStateChanged(recyclerView, newState);

if (scroll && (newState ==

RecyclerView.SCROLL\_STATE\_IDLE)) {

scrollRecyclerView();

}

rightLinkageLeft(newState);

}

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

super.onScrolled(recyclerView, dx, dy);

if (scroll) {

scrollRecyclerView();

}

}

});

vtl\_navigation.addOnTabSelectedListener(new VerticalTabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabView tabView, int i) {

isClickTab = true;

selectTag(i);

}

@Override

public void onTabReselected(TabView tabView, int i) {

}

});

}

/\*\*

\* 滑动 RecyclerView

\*/

private void scrollRecyclerView() {

scroll = false;

int indexDistance = index –

mLinearLayoutManager.findFirstVisibleItemPosition();

if (indexDistance >= 0 && indexDistance <

rv\_navigation.getChildCount()) {

int top =

rv\_navigation.getChildAt(indexDistance).getTop();

rv\_navigation.smoothScrollBy(0, top);

}

}

private void rightLinkageLeft(int newState) {

if (newState == RecyclerView.SCROLL\_STATE\_IDLE) {

if (isClickTab) {

isClickTab = false;

return;

}

int firstPosition = mLinearLayoutManager

.findFirstVisibleItemPosition();

if (index != firstPosition) {

index = firstPosition;

setChecked(index);

}

}

}

private void setChecked(int position) {

if (isClickTab) {

isClickTab = false;

} else {

if (vtl\_navigation == null) {

return;

}

vtl\_navigation.setTabSelected(index);

}

index = position;

}

private void selectTag(int i) {

index = i;

rv\_navigation.stopScroll();

smoothScrollToPosition(i);

}

private void smoothScrollToPosition(int currentPosition) {

int firstPosition = mLinearLayoutManager

.findFirstVisibleItemPosition();

int lastPosition = mLinearLayoutManager

.findLastVisibleItemPosition();

if (currentPosition <= firstPosition) {

rv\_navigation.smoothScrollToPosition(currentPosition);

} else if (currentPosition <= lastPosition) {

int top = rv\_navigation.getChildAt(currentPosition –

firstPosition).getTop();

rv\_navigation.smoothScrollBy(0, top);

} else {

rv\_navigation.smoothScrollToPosition(currentPosition);

scroll = true;

}

}

}

/\*\*

\* 学习导航布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal">

<q.rorbin.verticaltablayout.VerticalTabLayout

android:id="@+id/vtl\_navigation"

android:layout\_width="112dp"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

app:indicator\_color="@color/main\_white"

app:tab\_height="45dp"

app:tab\_margin="15dp"

app:tab\_mode="scrollable">

</q.rorbin.verticaltablayout.VerticalTabLayout>

<View

android:layout\_width="0.5dp"

android:layout\_height="match\_parent"

android:background="@color/Grey300" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_navigation"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v7.widget.RecyclerView>

</LinearLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 学习导航学习内容适配器

\*/

public class NavigationAdapter extends BaseQuickAdapter

<NavigationData, BaseViewHolder> {

public NavigationAdapter(int layoutResId, @Nullable

List<NavigationData> data) {

super(layoutResId, data);

}

@Override

protected void convert(BaseViewHolder helper, NavigationData item) {

if (!TextUtils.isEmpty(item.getName())) {

helper.setText(R.id.item\_navigation\_tv, item.getName());

}

TagFlowLayout mTagFlowLayout = helper.getView

(R.id.item\_navigation\_flow\_layout);

List<ArticleItemData> mArticles = item.getArticles();

mTagFlowLayout.setAdapter(new. TagAdapter<ArticleItemData>(mArticles) {

@Override

public View getView(FlowLayout parent, int position,

ArticleItemData articleItemData) {

TextView tv = (TextView) LayoutInflater

.from(parent.getContext())

.inflate(R.layout.tag\_flow\_layout\_tv,

mTagFlowLayout, false);

if (articleItemData == null) {

return null;

}

String name = articleItemData.getTitle();

tv.setText(name);

tv.setTextColor(getRandomColor());

mTagFlowLayout.setOnTagClickListener((view, position1,

parent1) -> {

Intent intent = new Intent(parent.getContext(),

ArticleDetailActivity.class); intent.putExtra(Constants.ARTICLE\_LINK,mArticles

.get(position).getLink()); intent.putExtra(Constants.ARTICLE\_TITLE,mArticles

.get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,mArticles

.get(position).getId());

intent.putExtra(Constants.IS\_COLLECTED,mArticles

.get(position).isCollect());

intent.putExtra(Constants.EVENT\_BUS\_TAG,

"flow\_layout");

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,-

1);

parent.getContext().startActivity(intent);

return true;

});

return tv;

}

});

}

/\*\*

\* 获取随机颜色

\* @return

\*/

public static int getRandomColor() {

Random random = new Random();

//0-190, 如果颜色值过大,就越接近白色,就看不清了,所以需要限定 范围

int red;

int green;

int blue;

red = random.nextInt(190);

green = random.nextInt(190);

blue = random.nextInt(190);

//使用rgb混合生成一种新的颜色,Color.rgb生成的是一个int数

return Color.rgb(red, green, blue);

}

}

/\*\*

\* 学习导航是适配器中布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/main\_white"

android:padding="12dp"

android:orientation="vertical">

<TextView

android:id="@+id/item\_navigation\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textColor="@color/black"

android:textSize="16sp"

tools:text="@string/app\_name" />

<com.zhy.view.flowlayout.TagFlowLayout

android:id="@+id/item\_navigation\_flow\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/main\_white"

android:padding="12dp"

app:max\_select="0" />

</LinearLayout>

public interface NavigationContract {

interface View extends BaseView {

void showNavigationData(List<NavigationData> navigationData);

}

interface Presenter extends BasePresenter<View> {

void getNavigationData();

}

}

/\*\*

\* 知识体系相关调用

\*/

public class KnowledgePresenter extends Presenter

<KnowledgeContract.View> implements KnowledgeContract.Presenter {

@Inject

public KnowledgePresenter() {

}

@Override

public void getKnowledgeSystemData(boolean isShowViewStatus) {

addSubscribe(dataManager.getKnowledgeData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(listBaseResponse -> mView != null)

.subscribeWith(new BaseObserver<List<KnowledgeSystemData>>(mView,

MobileLearnApp.getContext().getString

(R.string.failed\_to\_get\_knowledge),

isShowViewStatus) {

//知识体系数据请求成功后回调函数

@Override

public void success(List<KnowledgeSystemData>

knowledgeSystemData) {

mView.showKnowledgeSystemData(knowledgeSystemData);

}

}));

}

/\*\*

\* 重新加载

\*/

@Override

public void reload() {

getKnowledgeSystemData(false);

}

}

/\*\*

\* 知识体系主页面

\*/

public class KnowledgeFragment extends BaseFragment

<KnowledgePresenter> implements KnowledgeContract.View {

@BindView(R.id.rv\_knowledge)

RecyclerView rv\_knowledge;

@BindView(R.id.srl\_knowledge)

SmartRefreshLayout srl\_knowledge;

private KnowledgeAdapter mAdapter;

public static KnowledgeFragment newInstance() {

KnowledgeFragment knowledgeFragment = new KnowledgeFragment();

return knowledgeFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_knowledge;

}

@Override

protected void initView() {

ArrayList<KnowledgeSystemData> knowledgeSystemData = new

ArrayList<>();

mAdapter = new KnowledgeAdapter(R.layout.item\_knowledge,

knowledgeSystemData);

mAdapter.setOnItemClickListener((adapter, view, position) ->

startKnowledgeDetail(view, position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity,DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,R.drawable.shape\_hp\_rv\_line));

rv\_knowledge.addItemDecoration(divider);

rv\_knowledge.setHasFixedSize(true);

rv\_knowledge.setLayoutManager(new. LinearLayoutManager(\_mActivity));

rv\_knowledge.setAdapter(mAdapter);

srl\_knowledge.setOnRefreshListener(refreshLayout -> {

presenter.getKnowledgeSystemData(false);

refreshLayout.finishRefresh();

});

}

@Override

protected void initData() {

presenter.getKnowledgeSystemData(true);

}

@Override

public void showKnowledgeSystemData(List<KnowledgeSystemData>

knowledgeSystems) {

mAdapter.replaceData(knowledgeSystems);

}

/\*\*

\* 开启知识体系详情页

\* @param view

\* @param position

\*/

public void startKnowledgeDetail(View view, int position) {

Intent intent = new Intent(\_mActivity,

KnowledgeDetailActivity.class);

intent.putExtra(Constants.KNOWLEDGE\_DATA, mAdapter.getData().get(position));

startActivity(intent);

}

}

/\*\*

\* 知识体系主页面布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_knowledge"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_knowledge"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 知识体系详情页

\*/

public class KnowledgeDetailActivity extends BaseActivity

<KnowledgeDetailPresenter> implements KnowledgeDetailContract.View {

@BindView(R.id.toolbar)

Toolbar toolbar;

@BindView(R.id.toolbar\_title)

TextView toolbar\_title;

@BindView(R.id.tb\_knowledge\_detail)

TabLayout tb\_knowledge\_detail;

@BindView(R.id.vp\_knowledge\_detail)

ViewPager vp\_knowledge\_detail;

@BindView(R.id.toolbar\_back)

ImageView toolbar\_back;

private SparseArray<KnowledgeArticleFragment>

knowledgeArticleFragmentArray= new SparseArray<>();

private List<KnowledgeSystemData> dataChildren;

@Override

protected int getLayoutId() {

return R.layout.activity\_knowledge\_detail;

}

@Override

protected void initToolbar() {

setSupportActionBar(toolbar);

ActionBar actionBar = getSupportActionBar();

if (actionBar != null) {

actionBar.setDisplayShowTitleEnabled(false);

toolbar\_back.setVisibility(View.VISIBLE);

}

toolbar\_back.setOnClickListener(view -> finish());

getWindow().setNavigationBarColor(getResources().getColor(R.color.white\_smoke));

}

@Override

public void onBackPressedSupport() {

super.onBackPressedSupport();

}

@Override

protected void initView() {

}

/\*\*

\* 初始化点击事件和数据

\*/

@Override

protected void initEventAndData() {

KnowledgeSystemData knowledgeSystemData =

(KnowledgeSystemData). getIntent().getSerializableExtra(Constants.KNOWLEDGE\_DATA);

toolbar\_title.setText(knowledgeSystemData.getName());

dataChildren = knowledgeSystemData.getChildren();

vp\_knowledge\_detail.setAdapter(new. FragmentStatePagerAdapter(getSupportFragmentManager()) {

@Override

public Fragment getItem(int position) {

KnowledgeArticleFragment knowledgeArticleFragment =

knowledgeArticleFragmentArray.get(position);

if (knowledgeArticleFragment == null) {

Bundle bundle = new Bundle();

bundle.putInt("KnowledgeDataId", dataChildren.get(position).getId());

knowledgeArticleFragment =

KnowledgeArticleFragment.newInstance(bundle);

nowledgeArticleFragmentArray.put(position,knowledgeArticleFragment);

return knowledgeArticleFragment;

} else {

return knowledgeArticleFragment;

}

}

@Override

public int getCount() {

return dataChildren.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return Html.fromHtml(dataChildren

.get(position).getName());

}

@Override

public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {

System.out.println("destory");

}

});

vp\_knowledge\_detail.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tb\_knowledge\_detail));

tb\_knowledge\_detail.setupWithViewPager(vp\_knowledge\_detail);

tb\_knowledge\_detail.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(vp\_knowledge\_detail));

tb\_knowledge\_detail.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

vp\_knowledge\_detail.setCurrentItem(tab.getPosition()

,false);

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

@Override

public void onDestroy() {

if (knowledgeArticleFragmentArray != null) {

knowledgeArticleFragmentArray.clear();

knowledgeArticleFragmentArray = null;

}

if (dataChildren != null) {

dataChildren.clear();

dataChildren = null;

}

super.onDestroy();

}

}

/\*\*

\* 知识体系详情页布局文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<include layout="@layout/toolbar" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

android:orientation="vertical"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<android.support.design.widget.TabLayout

android:id="@+id/tb\_knowledge\_detail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/white"

app:tabIndicatorColor="@color/black"

app:tabIndicatorHeight="2dp"

app:tabMode="scrollable"

app:tabSelectedTextColor="@color/black"

app:tabTextColor="@color/Grey700" />

<android.support.v4.view.ViewPager

android:id="@+id/vp\_knowledge\_detail"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v4.view.ViewPager>

</LinearLayout>

</android.support.design.widget.CoordinatorLayout>

/\*\*

\* 公众号主页面

\*/

public class PublicNumFragment extends BaseFragment

<PublicNumPresenter> implements PublicNumContract.View {

@BindView(R.id.tb\_public\_num)

TabLayout tb\_public\_num;

@BindView(R.id.vp\_public\_num)

ViewPager vp\_public\_num;

private List<PublicNumData> mPublicNumList;

private SparseArray<PersonalPublicNumFragment> personalFragment =. new SparseArray<>();

private PersonalPublicNumFragment currentFragment;

public static PublicNumFragment newInstance() {

return new PublicNumFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_public\_num;

}

@Override

protected void initView() {

}

@Override

protected void initData() {

presenter.getPublicNumData();

}

/\*\*

\* 展示公众号列表

\* @param publicNumData 列表数据

\*/

@Override

public void showPublicNumData(List<PublicNumData> publicNumData) {

mPublicNumList = publicNumData;

vp\_public\_num.setAdapter(new FragmentStatePagerAdapter(getChildFragmentManager()) {

@Override

public Fragment getItem(int position) {

PersonalPublicNumFragment personalPublicNumFragment =

personalFragment.get(position);

if (personalPublicNumFragment == null) {

Bundle bundle = new Bundle();

bundle.putInt("publicNumId",

publicNumData.get(position).getId());

personalPublicNumFragment =

PersonalPublicNumFragment.newInstance(bundle);

personalFragment.put(position,personalPublicNumFragment);

return personalPublicNumFragment;

} else {

return personalPublicNumFragment;

}

}

@Override

public int getCount() {

return mPublicNumList == null ? 0 :

mPublicNumList.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return Html.fromHtml(mPublicNumList.get(position).getName());

}

//不添加会内存溢出

@Override

public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {

}

});

vp\_public\_num.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tb\_public\_num));

tb\_public\_num.setupWithViewPager(vp\_public\_num);

tb\_public\_num.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(vp\_public\_num));

tb\_public\_num.addOnTabSelectedListener(new. TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

vp\_public\_num.setCurrentItem(tab.getPosition(), false);

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

@Override

public void onDestroyView() {

super.onDestroyView();

if (mPublicNumList != null) {

mPublicNumList.clear();

mPublicNumList = null;

}

if (personalFragment != null) {

personalFragment.clear();

personalFragment = null;

}

}

}

/\*\*

\*公众号主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.design.widget.TabLayout

android:id="@+id/tb\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/main\_white"

app:tabTextColor="@color/bottom\_bar"

app:tabIndicatorColor="@color/Grey1000"

app:tabIndicatorHeight="2dp"

app:tabSelectedTextColor="@color/Grey900"

app:tabMode="scrollable"

/>

<android.support.v4.view.ViewPager

android:id="@+id/vp\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v4.view.ViewPager>

</LinearLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 个人公众号页面

\*/

public class PersonalPublicNumFragment extends BaseFragment

<PersonalPublicNumPresenter>

implements PersonalPublicNumContract.View {

@BindView(R.id.srl\_personal\_public\_num)

SmartRefreshLayout srl\_personal\_public\_num;

@BindView(R.id.rv\_public\_num)

RecyclerView rv\_public\_num;

private PersonalPublicNumAdapter mAdapter;

public static PersonalPublicNumFragment newInstance(Bundle bundle) {

PersonalPublicNumFragment personalPublicNumFragment = new

PersonalPublicNumFragment();

personalPublicNumFragment.setArguments(bundle);

return personalPublicNumFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_personal\_public\_num;

}

@Override

protected void initView() {

ArrayList<ArticleItemData> articleItemData = new ArrayList<>();

mAdapter = new PersonalPublicNumAdapter(R.layout.item\_article,

articleItemData);

mAdapter.setOnItemClickListener((adapter, view, position) ->

startArticleDetail(view，position));

mAdapter.setOnItemChildClickListener((adapter, view,

position). ->

childOnClick(view，position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity,

DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity, R.drawable.shape\_hp\_rv\_line));

rv\_public\_num.addItemDecoration(divider);

rv\_public\_num.setLayoutManager(new. LinearLayoutManager(\_mActivity));

//item 固定宽高，避免重复计算大小

rv\_public\_num.setHasFixedSize(true); rv\_public\_num.setAdapter(mAdapter);

}

@Override

protected void initData() {

int publicNumId = getArguments().getInt("publicNumId");

presenter.refreshLayout(publicNumId, true);

srl\_personal\_public\_num.setOnRefreshListener(refreshLayout ->。{

presenter.refreshLayout(publicNumId, false);

srl\_personal\_public\_num.finishRefresh();

});

srl\_personal\_public\_num.setOnLoadMoreListener(refreshLayout -> {

presenter.loadMore();

srl\_personal\_public\_num.finishLoadMore();

});

}

/\*\*

\* 打开学习资源详情

\* @param view

\* @param position

\*/

private void startArticleDetail(View view, int position) {

Intent intent = new Intent(\_mActivity,

ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK, mAdapter.getData().get(position).getLink());

intent.putExtra(Constants.ARTICLE\_TITLE, mAdapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID, mAdapter.getData().get(position).getId());

intent.putExtra(Constants.IS\_COLLECTED, mAdapter.getData().get(position).isCollect());

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON, true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION, position);

intent.putExtra(Constants.EVENT\_BUS\_TAG, "publicNum");

startActivity(intent);

}

private void childOnClick(View view, int position) {

if (presenter.getLoginStatus()) {

if (mAdapter.getData().get(position).isCollect()) {

presenter.cancelAddCollectArticle(position,

mAdapter.getData().get(position).getId());

} else {

presenter.addCollectArticle(position, mAdapter.getData().get(position).getId());

}

} else {

Intent intent = new Intent(\_mActivity,

LoginActivity.class);

startActivity(intent);

}

}

/\*\*

\* 展示公众号数据

\* @param articleListData 列表数据

\* @param isRefresh 是否刷新

\*/

@Override

public void showPersonalPublicNum(ArticleListData articleListData, boolean isRefresh) {

if (isRefresh) {

mAdapter.replaceData(articleListData.getDatas());

} else {

mAdapter.addData(articleListData.getDatas());

}

}

@Override

public void showCollectSuccess(int position) {

mAdapter.getData().get(position).setCollect(true);

mAdapter.setData(position, mAdapter.getData().get(position));

}

@Override

public void showCancelCollect(int position) {

mAdapter.getData().get(position).setCollect(false);

mAdapter.setData(position, mAdapter.getData().get(position));

}

}

/\*\*

\* 个人公众号布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_personal\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="?attr/colorPrimary">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_public\_num"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

android:visibility="visible" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 个人公众号适配器

\*/

public class PersonalPublicNumAdapter extends BaseQuickAdapter

<ArticleItemData, BaseViewHolder> {

public PersonalPublicNumAdapter(int layoutResId, @Nullable

List<ArticleItemData> data) {

super(layoutResId, data);

}

@Override

protected void convert(BaseViewHolder helper, ArticleItemData item) {

helper.setText(R.id.tv\_article\_title, item.getTitle()).setText(R.id.tv\_article\_author,

"作者：" + item.getAuthor())

.setImageResource(R.id.iv\_article\_like, item.isCollect() ? R.drawable.ic\_collect :

R.drawable.ic\_collect\_not);

if (!TextUtils.isEmpty(item.getNiceDate())) {

helper.setText(R.id.tv\_article\_public\_date, item.getNiceDate());

}

if (!TextUtils.isEmpty(item.getChapterName())) {

String classifyName = item.getSuperChapterName() + " / " + item.getChapterName();

helper.setText(R.id.tv\_article\_chapterName, classifyName);

}

//是否置顶

helper.getView(R.id.iv\_article\_top).setVisibility(item.getType() == 1 ? View.VISIBLE :View.GONE);

//是否是最新

helper.getView(R.id.iv\_article\_fresh).setVisibility(item.isFresh() ? View.VISIBLE :View.GONE);

if (item.getTags().size() > 0) {

helper.setText(R.id.tv\_article\_tag, item.getTags().get(0).getName()).

getView(R.id.tv\_article\_tag).setVisibility(View.VISIBLE);

} else {

helper.getView(R.id.tv\_article\_tag).setVisibility(View.GONE);

}

if (!TextUtils.isEmpty(item.getEnvelopePic())) {

helper.getView(R.id.iv\_article\_thumbnail).setVisibility(View.VISIBLE);

GlideUtil.download(mContext, item.getEnvelopePic(),

helper.getView(R.id.iv\_article\_thumbnail));

} else {

helper.getView(R.id.iv\_article\_thumbnail).setVisibility(View.GONE);

}

helper.addOnClickListener(R.id.iv\_article\_like);

}

}

/\*\*

\* 个人公众号适配器布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/item\_search\_pager\_group"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

app:cardBackgroundColor="@color/main\_white"

app:cardCornerRadius="0dp"

android:orientation="vertical"

app:cardElevation="1dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:background="@color/white"

android:paddingLeft="12dp">

<ImageView

android:id="@+id/iv\_article\_thumbnail"

android:layout\_width="130dp"

android:layout\_height="match\_parent"

android:layout\_gravity="center\_vertical"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="10dp"

android:scaleType="centerCrop"

android:visibility="gone"

tools:visibility="visible" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:paddingTop="10dp"

android:paddingRight="14dp"

android:paddingBottom="10dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="20dp"

>

<ImageView

android:id="@+id/iv\_article\_top"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginEnd="5dp"

android:src="@drawable/ic\_article\_top"

android:visibility="gone"

tools:visibility="visible"

/>

<ImageView

android:id="@+id/iv\_article\_fresh"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_marginEnd="5dp"

android:layout\_toEndOf="@id/iv\_article\_top"

android:layout\_toRightOf="@id/iv\_article\_top"

android:textColor="@color/red"

android:visibility="gone"

android:src="@drawable/ic\_article\_new"

tools:visibility="visible" />

<TextView

android:id="@+id/tv\_article\_tag"

android:layout\_width="wrap\_content"

android:layout\_height="15dp"

android:layout\_centerVertical="true"

android:layout\_marginEnd="5dp"

android:layout\_marginRight="5dp"

android:layout\_toEndOf="@id/iv\_article\_fresh"

android:layout\_toRightOf="@id/iv\_article\_fresh"

android:background="@drawable/background\_tag"

android:paddingLeft="4dp"

android:paddingTop="1dp"

android:paddingRight="4dp"

android:paddingBottom="2dp"

android:textColor="@color/colorPrimary"

android:textSize="10sp"

android:visibility="gone"

tools:text="wanandroid"

tools:visibility="visible" />

<TextView

android:id="@+id/tv\_article\_author"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_toEndOf="@id/tv\_article\_tag"

android:layout\_toRightOf="@id/tv\_article\_tag"

android:gravity="center"

android:textColor="@color/Grey700"

android:textSize="11sp"

tools:text="author" />

/>

<TextView

android:id="@+id/tv\_article\_public\_date"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_centerVertical="true"

android:textColor="@color/Grey500"

android:textSize="12sp" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp">

<TextView

android:id="@+id/tv\_article\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:ellipsize="end"

android:gravity="top|start"

android:lineSpacingExtra="2dp"

android:maxLines="2"

android:textColor="@color/black"

android:textSize="16sp"

tools:text="WanAndroid" />

</RelativeLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:gravity="center\_vertical">

<ImageView

android:id="@+id/iv\_article\_zan"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_zan"

android:visibility="gone"

/>

<ImageView

android:id="@+id/iv\_article\_like"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:contentDescription="@null"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<TextView

android:id="@+id/tv\_article\_chapterName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_alignParentRight="true"

android:gravity="center\_vertical"

android:textColor="@color/Grey700"

android:textSize="12sp"/>

</RelativeLayout>

</LinearLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

/\*\*

\* 公众号相关调用

\*/

public class PublicNumPresenter extends Presenter

<PublicNumContract.View> implements PublicNumContract.Presenter {

@Inject

public PublicNumPresenter() {

}

@Override

public void getPublicNumData() {

addSubscribe(dataManager.getPublicNumData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(publicNumList -> mView != null)

.subscribeWith(new BaseObserver<List<PublicNumData>>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_public\_num),true) {

@Override

public void success(List<PublicNumData> publicNumData) {

mView.showPublicNumData(publicNumData);

}

}));

}

}

/\*\*

\* 个人公众号相关调用

\*/

public class PersonalPublicNumPresenter extends

CollectPresenter<PersonalPublicNumContract.View>

implements PersonalPublicNumContract.Presenter {

private int currentPage = 1;

private boolean isRefresh = true;

private int id;

@Inject

public PersonalPublicNumPresenter() {

}

@Override

public void getPersonalPublicNum(boolean viewStatus) {

addSubscribe(dataManager.getPersonalPublicNumData(id, currentPage)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListDataBaseResponse -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to

\_get\_personal\_public\_num),viewStatus) {

@Override

public void success(ArticleListData listData) {

if(viewStatus && currentPage == 1 &&

listData.getDatas().size() < 1) {

mView.showEmpty();

} else {

mView.showPersonalPublicNum(listData,

isRefresh);

}

}

}));

}

/\*\*

\* 加载更多

\*/

@Override

public void loadMore() {

currentPage++;

isRefresh = false;

getPersonalPublicNum(false);

}

/\*\*

\* 刷新列表

\* @param id id

\* @param viewStatus 是否刷新

\*/

@Override

public void refreshLayout(int id, boolean viewStatus) {

currentPage = 1;

isRefresh = true;

this.id = id;

getPersonalPublicNum(viewStatus);

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

@Subscriber(tag = "publicNum")

public void getData(CollectEvent collectEvent){

if (mView == null) {

return;

}

if (collectEvent.isCancel()) {

mView.showCancelCollect(collectEvent.getArticlePosition());

} else {

mView.showCollectSuccess(collectEvent.getArticlePosition());

}

}

}

public interface PersonalPublicNumContract {

interface View extends CollectContract.View {

void showPersonalPublicNum(ArticleListData articleListData,

boolean isRefresh);

}

interface Presenter extends BasePresenter<View> {

void getPersonalPublicNum(boolean viewStatus);

void loadMore();

void refreshLayout(int id, boolean viewStatus);

}

}

/\*\*

\* 项目模块主页面

\*/

public class ProjectFragment extends BaseFragment<ProjectPresenter>

implements ProjectContract.View {

@BindView(R.id.tb\_project)

TabLayout tb\_project;

@BindView(R.id.vp\_project)

ViewPager vp\_project;

private SparseArray<ProjectListFragment> fragmentSparseArray = new

SparseArray<>();

private List<ProjectData> mProjectTreeData;

public static ProjectFragment newInstance() {

return new ProjectFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_project;

}

@Override

protected void initView() {

}

@Override

protected void initData() {

presenter.getProjectData();

}

/\*\*

\* 加载项目列表

\* @param projectData 列表数据

\*/

@Override

public void showProjectData(List<ProjectData> projectData) {

mProjectTreeData = projectData;

vp\_project.setAdapter(new

FragmentStatePagerAdapter(getChildFragmentManager()) {

@Override

public Fragment getItem(int position) {

ProjectListFragment projectListFragment =

fragmentSparseArray.get(position);

if (projectListFragment == null) {

Bundle bundle = new Bundle();

bundle.putInt("projectId",

mProjectTreeData.get(position).getId());

projectListFragment =

ProjectListFragment.newInstance(bundle);

fragmentSparseArray.put(position, projectListFragment);

return projectListFragment;

} else {

return projectListFragment;

}

}

@Override

public int getCount() {

return projectData.size();

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return

Html.fromHtml(mProjectTreeData.get(position).getName());

}

@Override

public void destroyItem(@NonNull ViewGroup container, int

position, @NonNull Object object) {

}

});

vp\_project.addOnPageChangeListener(new TabLayout.TabLayoutOnPageChangeListener(tb\_project));

tb\_project.setupWithViewPager(vp\_project);

tb\_project.addOnTabSelectedListener(new TabLayout.ViewPagerOnTabSelectedListener(vp\_project));

tb\_project.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {

@Override

public void onTabSelected(TabLayout.Tab tab) {

vp\_project.setCurrentItem(tab.getPosition(), false);

}

@Override

public void onTabUnselected(TabLayout.Tab tab) {

}

@Override

public void onTabReselected(TabLayout.Tab tab) {

}

});

}

@Override

public void onDestroyView() {

if (fragmentSparseArray != null) {

fragmentSparseArray.clear();

fragmentSparseArray = null;

}

if (mProjectTreeData != null) {

mProjectTreeData.clear();

mProjectTreeData = null;

}

super.onDestroyView();

}

}

/\*\*

\* 项目模块主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.design.widget.TabLayout

android:id="@+id/tb\_project"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/white"

app:tabGravity="center"

app:tabIndicatorColor="@color/Grey1000"

app:tabIndicatorHeight="2dp"

app:tabMode="scrollable"

app:tabSelectedTextColor="@color/Grey900"

app:tabTextColor="@color/Grey700">

</android.support.design.widget.TabLayout>

<android.support.v4.view.ViewPager

android:id="@+id/vp\_project"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

</android.support.v4.view.ViewPager>

</LinearLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 项目详情页

\*/

public class ProjectListFragment extends BaseFragment

<ProjectListFragmentPresenter> implements

ProjectListFragmentContract.View {

@BindView(R.id.srl\_project\_list)

SmartRefreshLayout srl\_project\_list;

@BindView(R.id.rv\_project\_list)

RecyclerView rv\_project\_list;

private ProjectListAdapter mAdapter;

public static ProjectListFragment newInstance(Bundle bundle) {

ProjectListFragment projectListFragment = new

ProjectListFragment();

projectListFragment.setArguments(bundle);

return projectListFragment;

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_project\_list;

}

@Override

protected void initView() {

List<ArticleItemData> mArticleList = new ArrayList<>();

mAdapter = new ProjectListAdapter(R.layout.item\_project\_list,

mArticleList);

mAdapter.setOnItemClickListener((adapter, view, position) ->

openArticleDetail(view, position));

mAdapter.setOnItemChildClickListener((adapter, view, position)

-> clickChild(view, position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity, DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,

R.drawable.shape\_hp\_rv\_line));

rv\_project\_list.addItemDecoration(divider);

rv\_project\_list.setLayoutManager(new LinearLayoutManager(\_mActivity));

rv\_project\_list.setHasFixedSize(true);

rv\_project\_list.setAdapter(mAdapter);

}

@Override

protected void initData() {

assert getArguments() != null;

int projectId = getArguments().getInt("projectId");

srl\_project\_list.setOnRefreshListener(refreshLayout -> {

presenter.refreshLayout(projectId, false);

refreshLayout.finishRefresh();

});

srl\_project\_list.setOnLoadMoreListener(refreshLayout -> {

presenter.loadMore();

refreshLayout.finishLoadMore();

});

presenter.refreshLayout(projectId,true);

}

private void openArticleDetail(View view,int position){

Intent intent = new Intent(\_mActivity,

ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,mAdapter.getData().get(position).getLink());

intent.putExtra(Constants.ARTICLE\_TITLE,mAdapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,mAdapter.getData().get(position).getId());

intent.putExtra(Constants.IS\_COLLECTED,mAdapter.getData().get(position).isCollect());

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,position);

intent.putExtra(Constants.EVENT\_BUS\_TAG,"project");

startActivity(intent);

}

private void clickChild(View view,int position){

if (presenter.getLoginStatus()) {

if (mAdapter.getData().get(position).isCollect()) {

presenter.cancelAddCollectArticle(position,mAdapter.getData().get(position).getId());

} else {

presenter.addCollectArticle(position,mAdapter.getData().get(position).getId());

}

} else {

Intent intent = new Intent(\_mActivity,

LoginActivity.class);

startActivity(intent);

}

}

@Override

public void showProjectListData(ArticleListData articleListData,

boolean isRefresh) {

if (isRefresh) {

mAdapter.replaceData(articleListData.getDatas());

} else {

mAdapter.addData(articleListData.getDatas());

}

}

@Override

public void showCollectSuccess(int position) {

mAdapter.getData().get(position).setCollect(true);

mAdapter.setData(position, mAdapter.getData().get(position));

}

@Override

public void showCancelCollect(int position) {

mAdapter.getData().get(position).setCollect(false);

mAdapter.setData(position, mAdapter.getData().get(position));

}

}

/\*\*

\*项目详情页布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_project\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="?attr/colorPrimary">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_project\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/white"

android:visibility="visible" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 项目列表适配器

\*/

public class ProjectListAdapter extends BaseQuickAdapter

<ArticleItemData, BaseViewHolder> {

public ProjectListAdapter(int layoutResId, @Nullable

List<ArticleItemData> data) {

super(layoutResId, data);

}

@Override

protected void convert(BaseViewHolder helper, ArticleItemData item) {

helper.setText(R.id.item\_project\_list\_title\_tv, Html.fromHtml(item.getTitle()))

.setText(R.id.item\_project\_list\_author\_tv,item.getAuthor())

.setImageResource(R.id.item\_project\_list\_like\_iv, item.isCollect() ? R.drawable.ic\_collect :

R.drawable.ic\_collect\_not);

if (!TextUtils.isEmpty(item.getDesc())) {

helper.setText(R.id.item\_project\_list\_content\_tv, Html.fromHtml(item.getDesc()));

}

if (!TextUtils.isEmpty(item.getNiceDate())) {

helper.setText(R.id.item\_project\_list\_time\_tv, item.getNiceDate());

}

if (!TextUtils.isEmpty(item.getEnvelopePic())) {

GlideUtil.download(mContext, item.getEnvelopePic(),

helper.getView(R.id.item\_project\_list\_iv));

}

helper.addOnClickListener(R.id.item\_project\_list\_like\_iv);

}

}

/\*\*

\* 项目列表适配器布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/item\_search\_pager\_group"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="0dp"

android:layout\_marginRight="0dp"

android:clickable="true"

android:focusable="true"

android:foreground="?android:attr/selectableItemBackground"

app:cardBackgroundColor="@color/white"

app:cardCornerRadius="0dp"

app:cardElevation="1dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="180dp"

android:orientation="horizontal"

android:paddingLeft="14dp"

android:paddingTop="10dp"

android:paddingRight="14dp"

android:paddingBottom="10dp">

<ImageView

android:id="@+id/item\_project\_list\_iv"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="2"

android:contentDescription="@null"

app:srcCompat="@mipmap/ic\_launcher" />

<RelativeLayout

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_marginStart="8dp"

android:layout\_marginTop="8dp"

android:layout\_marginEnd="5dp"

android:layout\_weight="5">

<TextView

android:id="@+id/item\_project\_list\_title\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="start"

android:ellipsize="end"

android:maxLines="1"

android:text="@string/app\_name"

android:textColor="@color/black"

android:textSize="16sp" />

<TextView

android:id="@+id/item\_project\_list\_content\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/item\_project\_list\_title\_tv"

android:layout\_marginTop="10dp"

android:ellipsize="end"

android:maxLines="4"

android:text="@string/app\_name"

android:textColor="@color/Grey800"

android:textSize="14sp" />

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:orientation="horizontal">

<ImageView

android:id="@+id/item\_project\_list\_like\_iv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_collect\_not" />

<TextView

android:id="@+id/item\_project\_list\_time\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_vertical"

android:layout\_marginStart="10dp"

android:textColor="@color/Grey700"

android:textSize="12sp"

tools:text="time" />

<RelativeLayout

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:gravity="center\_vertical">

<TextView

android:id="@+id/item\_project\_list\_author\_tv"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:text="@string/app\_name"

android:textColor="@color/Grey700"

android:textSize="12sp" />

</RelativeLayout>

</LinearLayout>

</RelativeLayout>

</LinearLayout>

</android.support.v7.widget.CardView>

/\*\*

\* 项目相关调用

\*/

public class ProjectPresenter extends Presenter

<ProjectContract.View> implements ProjectContract.Presenter {

@Inject

public ProjectPresenter() {

}

@Override

public void getProjectData() {

addSubscribe(dataManager.getProjectData()

.compose(RxJavaUtil.SchedulerTransformer())

.filter(projectTreeDataList -> mView != null)

.subscribeWith(new BaseObserver<List<ProjectData>>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_project), true) {

@Override

public void success(List<ProjectData> projectData) {

mView.showProjectData(projectData);

}

}));

}

@Override

public void reload() {

getProjectData();

}

}

/\*\*

\* 项目列表相关调用

\*/

public class ProjectListFragmentPresenter extends CollectPresenter

<ProjectListFragmentContract.View> implements

ProjectListFragmentContract.Presenter {

private int currentPage = 1;

private boolean isRefresh = true;

private int projectId;

@Inject

public ProjectListFragmentPresenter() {

}

@Override

public void refreshLayout(int projectId, boolean isShowStatusView) {

isRefresh = true;

currentPage = 1;

this.projectId = projectId;

getProjectListData(isShowStatusView);

}

@Override

public void getProjectListData(boolean isShowStatusView) {

addSubscribe(dataManager.getProjectListData(currentPage, projectId)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(articleListData -> mView != null)

.subscribeWith(new BaseObserver<ArticleListData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_get\_ project),isShowStatusView) {

@Override

public void success(ArticleListData

articleListData) {

if(isShowStatusView && currentPage == 1 &&

articleListData.getDatas().size() < 1) {

mView.showEmpty();

} else {

mView.showProjectListData(articleListData,

isRefresh);

}

}

}));

}

@Override

public void reload() {

refreshLayout(projectId, true);

}

@Override

public void loadMore() {

isRefresh = false;

currentPage++;

getProjectListData(false);

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

@Subscriber(tag = "project")

public void collectEvent(CollectEvent collectEvent) {

if (mView == null) return;

if (collectEvent.isCancel()) {

mView.showCancelCollect(collectEvent.getArticlePosition());

} else {

mView.showCollectSuccess(collectEvent.getArticlePosition());

}

}

}

/\*\*

\* 我的收藏主页面

\*/

public class CollectFragment extends BaseFragment

<CollectListPresenter> implements CollectListContract.View {

@BindView(R.id.srl\_collect\_list)

SmartRefreshLayout srl\_collect\_list;

@BindView(R.id.rv\_collect\_list)

RecyclerView rv\_collect\_list;

private List<ArticleItemData> mCollectList;

private CollectListAdapter mAdapter;

public static CollectFragment newInstance(){

return new CollectFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_collect\_list;

}

@Override

protected void initView() {

mCollectList = new ArrayList<>();

mAdapter = new CollectListAdapter(R.layout.item\_article,

mCollectList);

mAdapter.setOnItemClickListener((adapter, view, position) -> startArticleDetail(position));

mAdapter.setOnItemChildClickListener((adapter, view, position) -> collectArticle(position));

DividerItemDecoration divider = new

DividerItemDecoration(\_mActivity,DividerItemDecoration.VERTICAL);

divider.setDrawable(ContextCompat.getDrawable(\_mActivity,R.drawable.shape\_hp\_rv\_line));

rv\_collect\_list.addItemDecoration(divider);

rv\_collect\_list.setLayoutManager(new

LinearLayoutManager(\_mActivity));

rv\_collect\_list.setHasFixedSize(true);

rv\_collect\_list.setAdapter(mAdapter);

}

@Override

protected void initData() {

srl\_collect\_list.setOnRefreshListener(refreshLayout -> {

presenter.getCollectArticle(false);

refreshLayout.finishRefresh();

});

srl\_collect\_list.setOnLoadMoreListener(refreshLayout -> {

presenter.loadMore();

refreshLayout.finishLoadMore();

});

presenter.getCollectArticle(true);

}

private void startArticleDetail(int position){

Intent intent = new Intent(\_mActivity,

ArticleDetailActivity.class);

intent.putExtra(Constants.ARTICLE\_LINK,mAdapter.getData().get(position).getLink());

intent.putExtra(Constants.ARTICLE\_TITLE,mAdapter.getData().get(position).getTitle());

intent.putExtra(Constants.ARTICLE\_ID,mAdapter.getData().get(position).getOriginId());

intent.putExtra(Constants.IS\_COLLECTED,true);

intent.putExtra(Constants.IS\_SHOW\_COLLECT\_ICON,true);

intent.putExtra(Constants.ARTICLE\_ITEM\_POSITION,position);

intent.putExtra(Constants.EVENT\_BUS\_TAG,"collectList");

startActivity(intent);

}

private void collectArticle(int position){

presenter.cancelCollectInCollectPage(position, mAdapter.getData().get(position).getId(),

mAdapter.getData().get(position).getOriginId());

}

/\*\*

\* 展示收藏列表

\* @param articleListData 列表数据

\* @param isRefresh 是否刷新

\*/

@Override

public void showCollectList(ArticleListData articleListData,

boolean isRefresh) {

if (isRefresh) {

mAdapter.replaceData(articleListData.getDatas());

} else {

mAdapter.addData(articleListData.getDatas());

}

}

@Override

public void showCollectSuccess(int position) {

Toast.makeText(\_mActivity, getString(R.string.collect\_success),Toast.LENGTH\_SHORT).show();

}

@Override

public void showCancelCollect(int position) {

mAdapter.remove(position);

Toast.makeText(\_mActivity, getString(R.string.cancel\_collect),Toast.LENGTH\_SHORT).show();

}

}

/\*\*

\*我的收藏主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<com.classic.common.MultipleStatusView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/custom\_multiple\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<com.scwang.smartrefresh.layout.SmartRefreshLayout

android:id="@+id/srl\_collect\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:srlAccentColor="?attr/colorPrimary"

app:srlEnablePreviewInEditMode="true"

app:srlPrimaryColor="?attr/colorPrimary">

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

<android.support.v7.widget.RecyclerView

android:id="@+id/rv\_collect\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

android:visibility="visible" />

<com.scwang.smartrefresh.header.MaterialHeader

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.scwang.smartrefresh.layout.SmartRefreshLayout>

</com.classic.common.MultipleStatusView>

/\*\*

\* 登录主页面

\*/

public class LoginFragment extends BaseFragment<LoginFragmentPresenter> implements LoginFragmentContract.View {

@BindView(R.id.et\_username)

EditText et\_username;

@BindView(R.id.et\_password)

EditText et\_password;

@BindView(R.id.btn\_login)

TextView btn\_login;

@BindView(R.id.tv\_sign\_up)

TextView tv\_sign\_up;

private LoadingDialog dialog;

public static LoginFragment newInstance(){

return new LoginFragment();

}

@Override

public void loginSuccess() {

Toast.makeText(\_mActivity,getString(R.string.login\_success),Toast.LENGTH\_SHORT).show();

\_mActivity.finish();

}

@Override

public void registerSuccess(RegisterEvent registerEvent) {

et\_username.setText(registerEvent.getUsername());

et\_password.setText(registerEvent.getPassword());

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_login;

}

@Override

protected void initView() {

btn\_login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

login();

}

});

tv\_sign\_up.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

register();

}

});

}

/\*\*

\* 登录

\*/

private void login() {

String username = et\_username.getText().toString();

String password = et\_password.getText().toString();

if (TextUtils.isEmpty(username)) {

Toast.makeText(\_mActivity, getString(R.string.username\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else if (TextUtils.isEmpty(password)) {

Toast.makeText(\_mActivity, getString(R.string.password\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else {

presenter.login(username, password);

}

}

/\*\*

\* 注册

\*/

private void register(){

RegisterFragment registerFragment =

findFragment(RegisterFragment.class);

if (registerFragment == null) {

registerFragment = RegisterFragment.newInstance();

}

getSupportDelegate().showHideFragment(registerFragment, LoginFragment.this);

}

@Override

public void showLoading() {

if (dialog == null) {

dialog = new LoadingDialog(\_mActivity,

getString(R.string.logining), R.mipmap.ic\_dialog\_loading);

}

dialog.show();

}

@Override

public void hideLoading() {

if (dialog != null) {

dialog.dismiss();

dialog = null;

}

}

@Override

protected void initData() {

}

}

/\*\*

\* 登录主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/login\_scroll\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

android:orientation="vertical"

android:scrollbars="none"

android:visibility="visible"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="18dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="22dp"

android:text="@string/tv\_login\_title"

android:textColor="@color/Grey980"

android:textSize="28sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="35dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="8dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/et\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/username"

android:inputType="textEmailAddress"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="15dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="12dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/et\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/password"

android:inputType="textPassword"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<TextView

android:id="@+id/tv\_sign\_up"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="end"

android:layout\_marginRight="40dp"

android:text="@string/register"

android:textColor="#999999"

android:textSize="15sp" />

<TextView

android:id="@+id/btn\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="40dp"

android:layout\_marginRight="40dp"

android:background="@drawable/btn\_login\_bg"

android:gravity="center"

android:padding="12dp"

android:text="@string/login"

android:textColor="@color/white"

android:textSize="16sp" />

</LinearLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/login\_protocol"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="30dp"

android:textColor="#999999" />

</RelativeLayout>

</LinearLayout>

/\*\*

\* 登录页相关调用

\*/

public class LoginFragmentPresenter extends

Presenter<LoginFragmentContract.View>implements LoginFragmentContract.Presenter {

@Inject

public LoginFragmentPresenter() {

}

@Override

public void login(String username, String password) {

addSubscribe(dataManager.login(username, password)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(loginData -> mView != null)

.subscribeWith(new BaseObserver<LoginData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_login), true) {

@Override

public void success(LoginData loginData) {

setLoginStatus(true);

setLoginAccount(loginData.getUsername());

EventBus.getDefault().post(new LoginEvent());

mView.loginSuccess();

}

}));

}

@Override

public void registerEventBus() {

EventBus.getDefault().register(this);

}

@Override

public void unregisterEventBus() {

EventBus.getDefault().unregister(this);

}

@Subscriber()

public void RegisterSuccessEvent(RegisterEvent registerEvent) {

mView.registerSuccess(registerEvent);

}

}

public interface LoginFragmentContract {

interface View extends BaseView {

void loginSuccess();

void registerSuccess(RegisterEvent registerEvent);

}

interface Presenter extends BasePresenter<View> {

void login(String username,String password);

}

}

/\*\*

\* 注册主页面

\*/

public class RegisterFragment extends BaseFragment

<RegisterPresenter> implements RegisterContract.View {

@BindView(R.id.register\_username)

EditText register\_username;

@BindView(R.id.register\_password)

EditText register\_password;

@BindView(R.id.confirm\_password)

EditText confirm\_password;

@BindView(R.id.btn\_register)

TextView btn\_register;

@BindView(R.id.tv\_login)

TextView tv\_login;

private AlertDialog dialog;

public static RegisterFragment newInstance(){

return new RegisterFragment();

}

@Override

protected int getLayoutId() {

return R.layout.fragment\_register;

}

@Override

protected void initView() {

btn\_register.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

register();

}

});

tv\_login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

goLogin();

}

});

}

/\*\*

\* 注册

\*/

private void register() {

String username = register\_username.getText().toString();

String password = register\_password.getText().toString();

String password2 = confirm\_password.getText().toString();

if (TextUtils.isEmpty(username)) {

Toast.makeText(\_mActivity, getString(R.string.username\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else if (TextUtils.isEmpty(password)) {

Toast.makeText(\_mActivity, getString(R.string.password\_not\_empty),Toast.LENGTH\_SHORT)

.show();

} else if (TextUtils.isEmpty(password2)) {

Toast.makeText(\_mActivity, getString(R.string.confirm\_password\_not\_empty),Toast.LENGTH\_SHORT).show();

} else if (!TextUtils.equals(password, password2)) {

Toast.makeText(\_mActivity, getString(R.string.twice\_password\_not\_same),Toast.LENGTH\_SHORT).show();

} else {

presenter.register(username, password, password2);

}

}

@Override

protected void initData() {

}

@Override

public void registerSuccess() {

Toast.makeText(\_mActivity,getString(R.string.register\_success),Toast.LENGTH\_SHORT).show();

goLogin();

}

private void goLogin() {

LoginFragment loginFragment =

findFragment(LoginFragment.class);

if (loginFragment == null) {

loginFragment = LoginFragment.newInstance();

}

getSupportDelegate().showHideFragment(loginFragment, RegisterFragment.this);

}

@Override

public void showLoading() {

if (dialog == null) {

View view = LayoutInflater.from(\_mActivity)

.inflate(R.layout.login\_progress\_bar, null, false);

TextView loadingText =

view.findViewById(R.id.loading\_text);

loadingText.setText(getResources().getString(R.string.logining));

dialog = new AlertDialog.Builder(\_mActivity)

.setView(view).create();

Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawableResource(android.R.color.transparent);

}

dialog.show();

}

@Override

public void hideLoading() {

if (dialog != null) {

dialog.dismiss();

dialog = null;

}

}

}

/\*\*

\* 注册主页面布局

\*/

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:id="@+id/login\_scroll\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/main\_white"

android:orientation="vertical"

android:scrollbars="none"

android:visibility="visible"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="18dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="22dp"

android:text="@string/tv\_login\_title"

android:textColor="@color/Grey980"

android:textSize="28sp" />

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="35dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="8dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/register\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/username"

android:inputType="textEmailAddress"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="15dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="12dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/register\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/password"

android:inputType="textPassword"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="15dp"

android:layout\_marginRight="40dp"

android:layout\_marginBottom="12dp"

android:focusable="true"

android:focusableInTouchMode="true">

<EditText

android:id="@+id/confirm\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/confirm\_password"

android:inputType="textPassword"

android:textColor="@color/Grey980"

android:textSize="15sp" />

</android.support.design.widget.TextInputLayout>

<TextView

android:id="@+id/tv\_login"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="end"

android:layout\_marginRight="40dp"

android:text="@string/back\_login"

android:textColor="#999999"

android:textSize="15sp" />

<TextView

android:id="@+id/btn\_register"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="40dp"

android:layout\_marginTop="40dp"

android:layout\_marginRight="40dp"

android:background="@drawable/btn\_login\_bg"

android:gravity="center"

android:padding="12dp"

android:text="@string/confirm\_register"

android:textColor="@color/white"

android:textSize="16sp" />

</LinearLayout>

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/register\_protocol"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginBottom="30dp"

android:textColor="#999999" />

</RelativeLayout>

</LinearLayout>

/\*\*

\* 注册主页面相关调用

\*/

public class RegisterPresenter extends Presenter<RegisterContract.View> implements RegisterContract.Presenter {

@Inject

public RegisterPresenter() {

}

@Override

public void register(String username, String password,String

confirmPassword) {

addSubscribe(dataManager.register(username,password,confirmPassword)

.compose(RxJavaUtil.SchedulerTransformer())

.filter(loginDataBaseResponse -> mView != null)

.subscribeWith(new BaseObserver<LoginData>(mView,

MobileLearnApp.getContext().getString(R.string.failed\_to\_register), true) {

@Override

public void success(LoginData loginData) {

EventBus.getDefault().post(new 、

RegisterEvent(loginData.getUsername(),loginData.getPassword()));

mView.registerSuccess();

}

}));

}

}

public interface RegisterContract {

interface View extends BaseView {

void registerSuccess();

}

interface Presenter extends BasePresenter<View>{

void register(String username,String password,String

confirmPassword);

}

}

/\*\*

\* 自定义 Application

\*/

public class MobileLearnApp extends Application implements

HasActivityInjector{

private static Context context;

@Inject

DispatchingAndroidInjector<Activity> dispatchingAndroidInjector;

@Inject

public DataManager mDataManager;

private RefWatcher refWatcher;

@Override

public void onCreate() {

super.onCreate();

context = getApplicationContext();

DaggerAppComponent.builder()

.appModule(new AppModule(this))

.httpModule(new HttpModule())

.build().inject(this);

refWatcher = setLeakCanary();

}

/\*\*

\* 内存泄漏检测工具

\* @return 如果当前进程是给LeakCanary堆分析则返回

\*/

private RefWatcher setLeakCanary() {

if (LeakCanary.isInAnalyzerProcess(this)) {

return RefWatcher.DISABLED;

}

return LeakCanary.install(this);

}

public static RefWatcher getRefWatcher(Context context){

MobileLearnApp mobileLearnApp = (MobileLearnApp)

context.getApplicationContext();

return mobileLearnApp.refWatcher;

}

@Override

public AndroidInjector<Activity> activityInjector() {

return dispatchingAndroidInjector;

}

public static Context getContext(){

return context;

}

}

/\*\*

\* Module 告诉 Component 可以从这个类中获取依赖对象，生成各种类

\* Provides 用来生成类的实例

\* Name 存在相同依赖，告知 Component 到底提供哪一个依赖对象

\* Singleton 局部单例，可以保证注入的相应类中是单例

\* 需要参数的依赖对象，也要提供参数的方法

\*/

@Module

public class AppModule {

private final MobileLearnApp application;

public AppModule(MobileLearnApp application){

this.application = application;

}

@Provides

@Singleton

MobileLearnApp provideApplicationContext(){

return application;

}

@Provides

@Singleton

HttpHelper provideHttpHelper(HttpHelperImpl httpHelperImpl){

return httpHelperImpl;

}

@Provides

@Singleton

PreHelper providePreferenceHelper(PreHelperImpl preHelper){

return preHelper;

}

@Provides

@Singleton

DbHelper provideDbHelper(DbHelperImpl dbHelper){

return dbHelper;

}

@Provides

@Singleton

DataManager provideDataManager(HttpHelper httpHelper,PreHelper

preHelper,DbHelper dbHelper){

return new DataManager(httpHelper,preHelper,dbHelper);

}

}

@Module

public class HttpModule {

@Singleton

@Provides

ApiService provideApiService(Retrofit retrofit) {

return retrofit.create(ApiService.class);

}

@Singleton

@Provides

Retrofit provideRetrofit(Retrofit.Builder builder, OkHttpClient

client) {

return createRetrofit(builder, client, ApiService.BASE\_URL);

}

@Singleton

@Provides

Retrofit.Builder provideRetrofitBuilder() {

return new Retrofit.Builder();

}

//OkHttpClient 创建

@Singleton

@Provides

OkHttpClient.Builder provideOkHttpBuilder() {

return new OkHttpClient.Builder();

}

@Singleton

@Provides

OkHttpClient provideOkHttpClient(OkHttpClient.Builder builder) {

if (BuildConfig.DEBUG) {

//消息拦截器，可看到接口返回的所有内容 BASIC 级别是请求/响应行

HttpLoggingInterceptor loggingInterceptor = new

HttpLoggingInterceptor();

loggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BASIC);

builder.addInterceptor(loggingInterceptor);

}

//设置网络缓存

File cacheFile = new File(CacheUtils.getCacheDir());

System.out.println("HttpModule 设置缓存路径" + cacheFile);

Cache cache = new Cache(cacheFile, 1024 \* 1024 \* 100);//100M

builder.addNetworkInterceptor(new NetCacheInterceptor());

builder.cache(cache);

//设置超时时间

builder.connectTimeout(8, TimeUnit.SECONDS);

builder.readTimeout(15, TimeUnit.SECONDS);

builder.writeTimeout(15, TimeUnit.SECONDS);

//设置重新连接

builder.retryOnConnectionFailure(true);

//持久化 Cookie

builder.cookieJar(new PersistentCookieJar(new

SetCookieCache(),

new SharedPrefsCookiePersistor(MobileLearnApp.getContext())));

return builder.build();

}

private Retrofit createRetrofit(Retrofit.Builder builder,

OkHttpClient client, String baseUrl) {

return builder

.baseUrl(baseUrl)

.client(client)

.addCallAdapterFactory(RxJava2CallAdapterFactory.crea te())

.addConverterFactory(GsonConverterFactory.create()) //把retrofit的请求转化为RxJava

// 的Observable

.build();

}

}

public class HttpHelperImpl implements HttpHelper {

private ApiService mApiService;

@Inject

HttpHelperImpl(ApiService apiService) {

mApiService = apiService;

}

@Override

public Observable<BaseResponse<ArticleListData>>

getArticleList(int num) {

return mApiService.getArticleList(num);

}

@Override

public Observable<BaseResponse<ArticleListData>>

addArticleCollect(int id) {

return mApiService.addArticleCollect(id);

}

@Override

public Observable<BaseResponse<ArticleListData>>

cancelAddArticleCollect(int id) {

return mApiService.cancelAddArticleCollect(id);

}

@Override

public Observable<BaseResponse<List<ArticleItemData>>>

getUpArticles() {

return mApiService.getUpArticles();

}

@Override

public Observable<BaseResponse<List<BannerData>>> getBannerData() {

return mApiService.getBannerData();

}

@Override

public Observable<BaseResponse<List<PublicNumData>>>

getPublicNumData() {

return mApiService.getPublicNumData();

}

@Override

public Observable<BaseResponse<ArticleListData>>

getPersonalPublicNumData(int id,int page) {

return mApiService.getPersonalPublicNumData(id,page);

}

@Override

public Observable<BaseResponse<List<KnowledgeSystemData>>>

getKnowledgeData() {

return mApiService.getKnowledgeData();

}

@Override

public Observable<BaseResponse<List<ProjectData>>>

getProjectData() {

return mApiService.getProjectData();

}

@Override

public Observable<BaseResponse<ArticleListData>>

getProjectListData(int page, int projectId) {

return mApiService.getProjectListData(page,projectId);

}

@Override

public Observable<BaseResponse<List<NavigationData>>>

getNavigationData() {

return mApiService.getNavigationData();

}

@Override

public Observable<BaseResponse<LoginData>>

login(String username, String password) {

return mApiService.login(username,password);

}

@Override

public Observable<BaseResponse<LoginData>>

register(String username, String password,String confirmPassword) {

return mApiService.register(username,password,confirmPassword);

}

@Override

public Observable<BaseResponse<LoginData>> logout() {

return mApiService.logout();

}

@Override

public Observable<BaseResponse<ArticleListData>>

cancelCollectInCollectPage(int id,

int originId) {

return mApiService.cancelCollectInCollectPage(id,originId);

}

@Override

public Observable<BaseResponse<ArticleListData>>

getCollectList(int page) {

return mApiService.getCollectList(page);

}

@Override

public Observable<BaseResponse<List<WebsiteData>>>

getWebsiteList() {

return mApiService.getWebsiteList();

}

@Override

public Observable<BaseResponse<List<HotSearchData>>>

getSearchData() {

return mApiService.getSearchData();

}

@Override

public Observable<BaseResponse<ArticleListData>>

getSearchResultList(int pageNum, String key) {

return mApiService.getSearchResultList(pageNum,key);

}

/\*\*

\* 数据库实现类

\*/

public class DbHelperImpl implements DbHelper {

private final DaoMaster daoMaster;

private final DaoSession daoSession;

private String data;

private HistoryData historyData;

private List<HistoryData> historyDataList;

private static final int HISTORY\_LIST\_SIZE = 10;

@Inject

public DbHelperImpl() {

DaoMaster.DevOpenHelper devOpenHelper = new DaoMaster

.DevOpenHelper(MobileLearnApp.getContext(),Constants.DB\_NAM);

SQLiteDatabase database = devOpenHelper

.getWritableDatabase();

daoMaster = new DaoMaster(database);

daoSession = daoMaster.newSession();

}

/\*\*

\* 添加搜索历史

\* @param data

\* @return

\*/

@Override

public List<HistoryData> addHistoryData(String data) {

this.data = data;

getHistoryList();

createHistoryData();

if (historyDataForward()) {

return historyDataList;

}

if (historyDataList.size() < HISTORY\_LIST\_SIZE) {

getHistoryDataDao().insert(historyData);

} else {

historyDataList.remove(0);

historyDataList.add(historyData);

getHistoryDataDao().deleteAll();

getHistoryDataDao().insertInTx(historyDataList);

}

return historyDataList;

}

/\*\*

\* 清空搜索历史

\*/

@Override

public void clearHistoryData() {

daoSession.getHistoryDataDao().deleteAll();

}

/\*\*

\* 删除某一条搜索历史

\* @param id

\*/

@Override

public void deleteHistoryDataId(Long id) {

daoSession.getHistoryDataDao().deleteByKey(id);

}

/\*\*

\* 加载搜索历史

\* @return

\*/

@Override

public List<HistoryData> loadAllHistoryData() {

return daoSession.getHistoryDataDao().loadAll();

}

/\*\*

\* 获取历史搜索列表

\*/

private void getHistoryList() {

historyDataList = getHistoryDataDao().loadAll();

}

private void createHistoryData() {

historyData = new HistoryData();

historyData.setDate(System.currentTimeMillis());

historyData.setData(data);

}

private HistoryDataDao getHistoryDataDao() {

return daoSession.getHistoryDataDao();

}

/\*\*

\* 历史数据前移

\*

\* @return 返回true表示查询的数据已存在

\*/

private boolean historyDataForward() {

//重复搜索时进行历史记录前移

Iterator<HistoryData> iterator = historyDataList.iterator();

//不要在foreach循环中进行元素的remove、add操作

while (iterator.hasNext()) {

HistoryData historyData1 = iterator.next();

if (historyData1.getData().equals(data)) {

historyDataList.remove(historyData1);

historyDataList.add(historyData);

getHistoryDataDao().deleteAll();

getHistoryDataDao().insertInTx(historyDataList);

return true;

}

}

return false;

}

}

@Singleton

@Component(modules = {

AndroidInjectionModule.class, //支持四大组件都可以被绑定

AndroidSupportInjectionModule.class,

AllActivityModule.class,

AllFragmentModule.class,

AppModule.class,

HttpModule.class

})

public interface AppComponent {

void inject(MobileLearnApp mobileLearnApp);

}

/\*\*

\* 系统中用到的颜色代码

\*/

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="colorPrimary">#008577</color>

<color name="colorPrimaryDark">#00574B</color>

<color name="colorAccent">#28BFA0</color>

<color name="lightWhite">#cccccc</color>

<!--<color name="bottom\_select\_green">#08C261</color>-->

<color name="bottom\_select\_green">#28BFA0</color>

<color name="white">#FFFFFF</color>

<color name="white\_smoke">#F7F7F7</color>

<color name="Grey50">#FAFAFA</color>

<color name="Grey100">#F5F5F5</color>

<color name="Grey200">#EEEEEE</color>

<color name="Grey250">#EDEDED</color>

<color name="Grey280">#E2E2E2</color>

<color name="Grey300">#E0E0E0</color>

<color name="Grey350">#D9D9D9</color>

<color name="Grey400">#bdbdbd</color>

<color name="Grey500">#9e9e9e</color>

<color name="Grey600">#757575</color>

<color name="Grey700">#616161</color>

<color name="Grey750">#363636</color>

<color name="Grey800">#3D3D3D</color>

<color name="Grey900">#212121</color>

<color name="Grey950">#181818</color>

<color name="Grey980">#333333</color>

<color name="Grey1000">#1A1A1A</color>

<color name="black">#19191B</color>

<color name="black\_pure">#000000</color>

<color name="navigation\_item\_ic\_color">#999999</color>

<color name="red">#f44336</color>

<color name="navigation\_bar">#F0F0F0</color>

<color name="bottom\_bar">#999999</color>

<color name="bottom\_bar\_bg">#FFFFFF</color>

<color name="main\_white">#FDFDFD</color>

<color name="green">#28BFA0</color>

</resources>

/\*\*

\* SharedPreferences 实现类

\*/

public class PreHelperImpl implements PreHelper {

private final SharedPreferences sp;

@Inject

PreHelperImpl(){

MobileLearnApp.getContext().getSharedPreferences(Constants.MY\_SP, Context.MODE\_PRIVATE);

}

@Override

public void setLoginStatus(boolean isLogin) {

sp.edit().putBoolean(Constants.LOGIN\_STATUS,isLogin).apply();

}

@Override

public boolean getLoginStatus() {

return sp.getBoolean(Constants.LOGIN\_STATUS,false);

}

@Override

public void setLoginAccount(String account) {

sp.edit().putString(Constants.ACCOUNT,account).apply();

}

@Override

public String getLoginAccount() {

return sp.getString(Constants.ACCOUNT,"");

}

@Override

public void setNightMode(boolean isNightMode) {

}

@Override

public boolean isNightMode() {

return false;

}

}

public interface PreHelper {

void setLoginStatus(boolean isLogin);

boolean getLoginStatus();

void setLoginAccount(String account);

String getLoginAccount();

void setNightMode(boolean isNightMode);

boolean isNightMode();

}

//系统底部按钮布局

<?xml version="1.0" encoding="utf-8"?>

<menu xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto">

<item

android:id="@+id/tab\_home\_pager"

android:enabled="true"

android:icon="@drawable/ic\_home\_pager"

android:title="@string/home\_page"

app:showAsAction="ifRoom"> <!--如果有足够的空间会显示-->

</item>

<item

android:id="@+id/tab\_navigation"

android:enabled="true"

android:title="@string/navigation"

android:icon="@drawable/ic\_navigation"

app:showAsAction="ifRoom">

</item>

<item

android:id="@+id/tab\_knowledge"

android:enabled="true"

android:title="@string/knowledge"

android:icon="@drawable/ic\_knowledge"

app:showAsAction="ifRoom">

</item>

<item

android:id="@+id/tab\_public\_num"

android:enabled="true"

android:title="@string/public\_num"

android:icon="@drawable/ic\_public\_num"

app:showAsAction="ifRoom">

</item>

<item

android:id="@+id/tab\_project"

android:enabled="true"

android:title="@string/project"

android:icon="@drawable/ic\_project"

app:showAsAction="ifRoom">

</item>

</menu>

/\*\*

\* 系统清单文件

\*/

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.sunshinexu.mobilelearn">

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission

android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission

android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

<application

android:name=".app.MobileLearnApp"

android:allowBackup="true"

android:icon="@drawable/ic\_logo1"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name=".activity.main.ui.SplashActivity"

android:theme="@style/SplashTheme">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<action android:name="android.intent.action.VIEW" />

<category

android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".MainActivity" />

<activity android:name=".activity.main.ui.LoginActivity"/>

<activity. android:name=".activity.main.ui.ArticleDetailActivity" />

<activity

android:name=".activity.main.ui.KnowledgeDetailActivity"/>

<activity android:name=".activity.main.ui.ItemActivity"/>

<activity android:name=".activity.main.ui.SearchActivity"/>

</application>

</manifest>

# 参考文献

1. [1] 乔汉文.中小学生移动学习现状调查与对策研究[D].曲阜师范大学,2016. [↑](#endnote-ref-2)
2. [2] 王龙.平板电脑在实践教学中的应用[J].卷宗,2016,(4):172-173. [↑](#endnote-ref-3)
3. [3] CNNIC:中国互联网络信息中心发布第43次《中国互联网络发展状况统计报

   告》[J].中国互联网络信息中心,2019,(122):13. [↑](#endnote-ref-4)
4. [4] 侯妍.基于Android的计算机网络课程学习系统设计与实现[D].西安:西安

   石油大学,2017. [↑](#endnote-ref-5)
5. [5] 杨红云,覃李静,黎晨.近年来国外移动学习发展现状研究[J].软件导刊:教

   育技术,2016,15(3):91-93. [↑](#endnote-ref-6)
6. [6] 陈婷.基于Android的移动学习系统研发[D].广州:华南理工大学,2015. [↑](#endnote-ref-7)
7. [7] 白宇宇.基于Android的移动学习交互平台设计[D].北京:北京交通大学,

   2015. [↑](#endnote-ref-8)
8. [8] 王东琴.基于Android的电动汽车充电桩移动平台客户端开发[D].合肥:合肥工业大学,2017. [↑](#endnote-ref-9)
9. [9] 陶慧,吕爱华.Android移动客户端同步手绘板的设计与实现[J].工业和信

   息化教育,2017,(12):74-77. [↑](#endnote-ref-10)
10. [10]李群.基于OkHttp的文件传输设计与实现[J].电子技术与软件工程,2018,

    (13):180-181. [↑](#endnote-ref-11)
11. [11]刘望舒.Android进阶之光[M].北京:电子工业出版社,2017. [↑](#endnote-ref-12)
12. [12]姚世明.JSON在Android和Web通信中的应用研究[J].通信世界2015,

    (23):46-47. [↑](#endnote-ref-13)
13. [13]李乔.基于Android的在线学习系统的设计与实现[D].吉林:吉林大学，2018. [↑](#endnote-ref-14)
14. [14]任玉刚.Android开发探索艺术[M].北京:电子工业出版社,2015. [↑](#endnote-ref-15)
15. [15]郭霖.第一行代码第2版[M].北京:人民邮电出版,2016. [↑](#endnote-ref-16)