

倚动® 实验室

互联网+创新创业实战系列教材



全栈App 工程师

实战案例教程

主编 钟元生 赵圣鲁



Full App-Stack Developer

 江西高校出版社
JIANGXI UNIVERSITIES AND COLLEGES PRESS

互联网+创新创业实战系列教材

App 全栈工程师 实战案例教程

钟元生 赵圣鲁 主编

江西高校出版社

二〇一七年七月

图书在版编目 (CIP)数据

App 全栈工程师实战案例教程/钟元生, 赵圣鲁主编.
—南昌: 江西高校出版社, 2017.8
ISBN 978-7-5493-5914-1

I .①A... II.①钟...②赵...III.①移动终端—应用程序—程序设计—教材 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 195411 号

出版发行	江西高校出版社
社 址	江西省南昌市洪都北大道 96 号
邮 政 编 码	330046
总编室电话	(0791)88504319
销 售 电 话	(0791)88522272
网 址	www. juacp. com
印 刷	南昌市光华印刷有限责任公司
经 销	各地新华书店
开 本	787 mm x 1092 mm 1/16
印 张	32
字 数	700 千字
版 次	2017 年 8 月第 1 版第 1 次印刷
印 数	0001~2000 册
书 号	ISBN 978-7-5493-5914-1
定 价	78.90 元

赣版权登字-07-2017-917
版权所有 侵权必究

内容简介

围绕一个真实项目，通过一个个小案例，引导读者短时间内熟悉一个综合 App 的开发，以培养 App 程序员的独立开发能力。

全书包括 App 项目体验、需求分析、App 全栈开发微案例、无对话框的单页界面设计、含对话框单页界面设计、多页面切换效果设计、绘制趋势图、使用工具管理 SQLite 数据库、SQLite 数据库的操作模块设计、使用 MySQL 管理网络数据库、App 网络服务器端设计、App 客户端与服务器交互设计、App 集成测试和 App 全栈工程师创业思维训练。

本书适合作为“App 开发”实训课程教材，可供项目经验少的学生开发实用 App 时参考，也可作为大学生“互联网+”创新创业竞赛的实战指导书。



微信公众号：yd-lab

实验室网址：www.xs360.cn

Q Q 群：23169 3334

倚动图书网：www.xs360.cn/book

作者简介



钟元生，1968 年生，江西财经大学教授，博士生导师，浙江大学博士，美国加州大学尔湾分校访问学者，江西省人民政府学位委员会学第一届、二届学科评议组成员，江西省普通高校本科教学指导委员会计算机类专家委员，江西省物联网与大数据工程技术研究中心主任，软件与通信工程学院学术委员会主任、教育技术学硕士点研究生导师组组长、教育技术研究所所长。参与创办江西财经大学软件学院，曾任计算机应用技术、软件工程等专业硕士生导师，学校本科水平评估工作评建创优专家组副组长；国家职业技能鉴定专家委员会电子商务专业委员会委员，中国电子学会初级电子信息工程师职业资格认证工作委员会成员；省计算机学会理事；江西浙大校友会理事，浙江大学江财校友会副会长；科技部科技支撑计划会议评审专家、两届江西省研究生学位点评审专家、一届江西省教学成果奖评审专家；教育部主管、清华大学主办的《现代教育技术》期刊审稿专家；江西省教育厅大学生手机软件设计赛发起人、总策划；获江西财经大学“十大优秀研究生导师”和“教学十佳”等称号。

主持国家自然科学基金 1 项、副主持 2 项；完成省级以上课题 9 项。获省级优秀教学成果奖 4 项（2 项第一主持且获一等奖），出版专著 2 部，主编教材多部在清华大学出版社、电子工业出版社等处出版。



赵圣鲁，1990 年生，江西信息应用职业技术学院教师，江西财经大学软件与通信工程学院教育技术学硕士毕业，研究方向为移动学习与手机软件开发技术方向，专攻手机软件开发人才培养，具有多年的 App 开发和教学经验。参编教材《App 开发案例教程》（2015 年清华大学出版社出版）。

前言

“互联网+”创业公司中，App 是标配。现有教材多数重点讲述服务器端、客户端以及数据库等之一部分，对应于软件项目组中服务器程序员、Android/iOS 程序员、数据库程序员等岗位，比较适合有一定规模的公司。目前许多培训机构以及高校都是按某一类岗位的要求来培养程序员的，要找到这类程序员更容易。

很多时候，企业希望程序员同时承担设计服务器端、客户端和数据库等多个岗位的任务，即有全栈工程师的能力，初创公司尤其如此。即使在程序员分工较细的公司，如果程序员具有全栈工程师的基本能力，不同岗位的程序员更易合作。

随着全国大学生“互联网+”创新创业竞赛的不断推进，许多学生都想将自己的好点子用 App 实现，往往就卡在最终 App 的实现上，创意落地难。

教材为满足全栈工程师的培养要求，既要有一个完整的综合案例，又要能将案例拆分为一个个可独立运行、可测试的小例子。读者边学边做，边检验自己的知识掌握程度，学完全书即完成完整项目。这类教材设计比较难，市面少见，本书尝试之。

本书案例选自早期“豹考通”App 的简化版，介绍了该 App 的 Android 客户端程序、服务器端数据库、服务器管理程序开发以及 App 项目集成测试的全过程，并特设了“互联网+”创业思维训练内容，使全栈工程师培养更有价值。假定读者有一定的编程语言基础，对 App 开发和 Android 编程有一定的了解。

案例的有些代码较长，有些模块本书只列出关键代码，基于这些关键的代码讲解，读者可以访问教学资料的网站下载完整代码，边阅读代码边体验程序效果，并扩充或修改程序功能。为方便阅读，书中的每段代码都引入了代码编号，部分关键语句加了注释，并给出程序在资源包中的位置。

程序清单：Code0401\app\src\main\java\cn\jxufe\iet\code0401\ControlLineActivity.java

```
1 private class MyItemSelectedListener implements
   AdapterView.OnItemSelectedListener {
2     @Override
3     public void onItemSelected(AdapterView<?> parent, View view,
4                               int position, long id) {
5         switch (parent.getId()) {
6             case R.id.areaSpinner:// 如果是选择省份列表
7                 sourceAreaId = position;
8                 break;
9             case R.id.yearSpinner:// 如果是选择年份列表
10                .....
11            }
12        }
13    }
14    /*设置标题显示*/
15    controllLineTitle.setText(Html.fromHtml("<font color=red><b>"
16    .....
17    )
```

其中 1、2、3.....为代码行号，中间为代码实际内容，//后为注释内容。程序清单的位置也详细标注，便于读者查找下载。

在您阅读本书时，最好按照书中的步骤同时实际操作，加深印象，掌握关键技术，不建议仅直接打开源码运行、查看最后效果。编程基础较弱者，也可泛读全书、体验 App 开发流程和关键细节。尽量多在涉及相关技术时，参考相关教材或网络资源。

本书作为一种尝试，我们反复对案例选择和教材结构、教法、编法等进行研究与设计，历经一年半，撰写了几十个版本，进行了三次大规模修改，方得以展现在读者面前。尽管如此，依然有许多不尽人意的地方，如尚未完全实现方便的模块设计和集成测试，课后练习不够完善，学习效果过程检查点偏少。

为方便学习、交流与资源共享，我们提供了本书相关资源的下载地址，网址为：
<http://www.xs360.cn/book>。

全书由钟元生担任主编，负责全书的方案设计、内容策划、案例分解、细节把握、质量控制和统编定稿工作。各章分工如下：钟元生完成第 0 章、第 2 章并与其他作者合作完成其余各章的撰写，赵圣鲁负责第 1 章、第 4 章、第 5 章、第 12 章，钟坚负责第 3 章、第 11 章，邹宇杰负责第 6 章、第 7 章，吴凯负责第 8 章、第 9 章、第 10 章，涂露露负责第 13 章，廖琪玲负责第 14 章。高成珍、陈海俊、曹权为“豹考通”教学版的前期开发作了重要贡献。

在此，特别感谢支持本书编写出版工作的江西财经大学软件与通信工程学院、江西高校出版社有关领导和老师的帮助。

希望本书的出版有助于“App 开发实训”课程的教师备课；有助于教师更好地指导移动互联网软件方面的毕业设计，提高指导效率。不当之处，还望大家批评指正，以便将来再版时完善。

本书适合作为本专科“App 开发”课程实训教材，供无项目经验学生开发 App 时参考，可作为大学生“互联网+”创新创业竞赛的实战参考书。

编者

于江西财经大学麦庐园

2017 年 6 月

目 录

第 0 章 导读	1
0.1 开发概述	1
0.2 App 开发流程	1
0.3 案例介绍	5
0.4 本书教学中所用到的学习案例代码	6
0.5 配置局域网运行和开发环境	7
0.6 本书内容简介	8
0.7 课后练习	8
第 1 章 App 应用体验	9
1.1 “豹考通”项目介绍	9
1.2 开发环境配置	9
1.2.1 Java 语言	9
1.2.2 Java 环境配置	10
1.3 Android Studio 下载和安装	20
1.3.1 Android Studio 下载	20
1.3.2 Android Studio 安装	20
1.3.3 创建第一个项目：HelloWorld	26
1.3.4 创建一个虚拟机设备	29
1.4 “豹考通”服务端集成包运行	33
1.5 “豹考通”Android 端程序导入运行	35
1.5.1 项目导入	35
1.5.2 客户端运行效果图	38
1.5.3 “豹考通”项目客户端结构图	39
1.6 课后练习	42
第 2 章 项目需求分析	43
2.1 功能需求分析	43
2.1.1 需求获取	43
2.1.2 功能分析	43
2.1.3 系统 UML 建模	44
2.2 系统用例分析	45

2.2.1 系统用例图.....	45
2.2.2 系统用例描述	45
2.3 项目数据库分析	50
2.3.1 数据库基本概念	50
2.3.2 数据库设计的基本过程.....	50
2.3.3 系统所涉的实体及其属性	51
2.4 课后练习	54
第 3 章 App 全栈开发微案例	55
3.1 微案例介绍	55
3.2 本地数据版 App 案例	56
3.2.1 Android 界面设计	56
3.2.2 Android 功能实现	60
3.3 网络数据版 App 案例	63
3.3.1 建立 MySQL 数据库	63
3.3.2 设计网络服务器	67
3.3.2.1 建立数据库连接类.....	67
3.3.2.2 建立省控线实体类.....	73
3.3.2.3 数据库操作类的实现.....	74
3.3.2.4 Servlet 类的实现.....	76
3.3.3 Android 功能实现	82
3.3.3.1 建立实体类	82
3.3.3.2 建立服务器访问工具类.....	84
3.3.3.3 修改 demo.java Activity 文件.....	88
3.4 课后练习	90
第 4 章 无对话框的单页面设计	91
4.1 本章案例实现流程.....	91
4.2 “历年省控线查询”模块实现.....	93
4.2.1 界面设计.....	93
4.2.2 Spinner 实现下拉列表选项	98
4.2.3 事件监听器	101
4.2.4 ListView 列表	102
4.3 “历年录取线查询”模块实现.....	107
4.3.1 界面设计.....	107
4.3.2 历年录取线筛选条件栏实现	116
4.3.3 事件监听器	119

4.3.4 ListView 列表.....	123
4.4 课后练习	134
第 5 章 含对话框单页面界面设计	137
5.1 本章案例实现流程.....	137
5.2 “报考咨询”模块实现.....	139
5.2.1 界面设计.....	139
5.2.2 “问题列表”的实现	143
5.2.3 “问题回复”对话框实现	148
5.3 “个人基本信息”模块实现.....	149
5.3.1 界面设计.....	149
5.3.2 SharedPreferences 实现个人信息存储	153
5.3.3 “个人基本信息填写对话框”模块实现.....	155
5.4 课后练习	163
第 6 章 多页面切换效果设计	165
6.1 本章案例流程	165
6.2 多页面切换效果了解	166
6.3 页面切换框架实现.....	167
6.3.1 项目结构.....	167
6.3.2 Tabhost 界面布局文件设计	170
6.3.3 为选项卡添加样式	174
6.3.4 初始化选项卡设计	176
6.4 为选项卡对应添加内容	179
6.4.1 创建空 fragment 布局文件及其加载类	179
6.4.2 事件监听实现多页面切换	185
6.5 多页面内容填充.....	188
6.5.1 替换 page1.xml 页面为“省控线页面”	188
6.5.1.1 替换 xml 布局文件	189
6.5.1.2 替换 java 类.....	192
6.5.2 替换 page2 页面为“录取线查询”页面	208
6.5.2.1 替换 xml 布局文件	208
6.5.2.2 替换 java 类.....	213
6.6 课后练习	224
第 7 章 绘制趋势图.....	225
7.1 本章案例流程.....	225
7.2 绘图功能了解.....	226

7.3 项目结构	227
7.4 趋势图页面框架实现	229
7.4.1 趋势图 java 类的建立	229
7.4.2 在 xml 文件中引用自定义的趋势图控件	232
7.5 用绘图类绘制坐标轴	234
7.5.1 使用 Paint 类设置画笔	234
7.5.2 创建 onDraw()方法	235
7.5.3 绘制趋势图的坐标轴	236
7.5.3.1 绘制 x 轴、y 轴直线	236
7.5.3.2 绘制坐标轴的箭头	239
7.5.3.3 绘制坐标轴上的文字	241
7.6 用绘图类绘制趋势线—源数据来自数组	244
7.6.1 创建模拟数据	244
7.6.2 根据源数据绘制信息点	247
7.6.2.1 绘制 x 轴上的信息	247
7.6.2.2 根据源数据绘制趋势图	253
7.7 用绘图类绘制趋势线—源数据来自录取线页面	269
7.7.1 编写按钮点击事件和业务逻辑	269
7.7.1.1 基本实现过程	269
7.7.1.2 使用 Android Studio 导入 Gson 包	272
7.7.2 编写趋势图相关 Activity 和 Fragment	274
7.7.2.1 编写趋势图 Fragment 类	275
7.7.2.2 编写趋势图的 Activity	279
7.7.2.3 为 ResultActivity 进行注册	281
7.8 课后练习	283
第 8 章 使用工具管理 SQLite 数据库	285
8.1 了解 SQLite	287
8.2 SQLite 优点	287
8.3 SQLite 使用入门	289
8.3.1 安装使用 SQLite	289
8.3.2 SQLite 常用操作	292
8.4 创建 SQLite 数据库--bkt	294
8.4.1 创建数据库	294
8.4.2 创建地区表(area)	295
8.4.3 插入地区表(area)数据	296

8.4.4 创建学校表(school).....	297
8.4.5 导入学校表(school)数据	298
8.5 课后练习	304
第 9 章 SQLite 数据库的操作模块设计	305
9.1 本章案例流程	305
9.2 用 SQL 代码创建数据库及表	306
9.2.1 项目结构.....	306
9.2.2 案例实现.....	306
9.3 SQLite 数据库存储	315
9.3.1 SQLiteDatabase 类	316
9.3.2 SQLiteOpenHelper 类	316
9.3.3 Cursor 接口	317
9.3.4 ContentValues 类	317
9.4 使用 SQL 语句操作数据库	318
9.5 从 SQLite 获取 ListView 列表项的值	319
9.5.1 项目结构.....	319
9.5.2 案例流程模块	319
9.5.3 定义全局变量类	320
9.5.4 数据库操作辅助类	321
9.5.5 替换地区(area)表数据	326
9.5.6 替换学校(school)表数据	329
9.6 课后练习	331
第 10 章 使用 MySQL 管理网络数据库	333
10.1 MySQL 数据库的构建	333
10.1.1 MySQL 的应用范围	333
10.1.2 MySQL 的优缺点	333
10.1.3 MySQL 安装	334
10.2 数据库可视化工具安装和介绍	339
10.2.1 Navicat for MySQL 下载	339
10.2.2 Navicat for MySQL 安装	341
10.3 “豹考通”项目数据分析与设计	344
10.4 数据库表设计	344
10.4.1 各表的结构设计	344
10.4.2 建库、建表操作	350
10.4.3 使用 SQL 语句建表	353

10.4.4 SQL 语句讲解	357
10.4.5 制作推荐报告举例	360
10.5 项目数据导入与导出	364
10.5.1 Excel 表格导入数据库表	364
10.5.2 SQL 文件导入库	369
10.5.3 数据导出	371
10.6 课后练习	375
第 11 章 App 网络服务器管理程序设计	377
11.1 服务器开发背景知识	377
11.1.1 JSP 简介	377
11.1.2 Tomcat 服务器	377
11.1.3 服务器与客户端交互图解	382
11.2 了解 Java Web 技术	383
11.2.1 DAO 设计模式	383
11.2.2 认识 Java Web 程序的目录结构	384
11.3 Java Web 核心技术	385
11.3.1 JavaBean 技术	385
11.3.2 运行你的第一个 Java Web 程序	385
11.3.3 Servlet 技术	388
11.3.4 掌握 HttpServletRequest 类	389
11.3.5 掌握 HttpServletResponse 类	390
11.3.6 实战演练	390
11.4 设计 App 服务器数据库工具类	392
11.4.1 JDBC 技术	393
11.4.2 数据库连接类的实现	393
11.5 设计 App 服务器业务逻辑类	397
11.5.1 建立实体类	397
11.5.2 数据库操作类的实现	399
11.5.3 练习	406
11.6 设计 App 服务器 Servlet 类	407
11.6.1 省控线 Servlet 类的实现	407
11.6.2 练习	412
11.7 设计“豹考通”服务器问题模块	412
11.7.1 建立问题模块实体类	412
11.7.2 数据库问题模块操作类	414

11.7.3 问题模块 Servlet 类.....	420
11.8 课后练习	422
第 12 章 App 客户端与服务器交互设计.....	423
12.1 客户端和服务端数据交互基础.....	423
12.1.1 HttpClient.....	423
12.1.2 网络服务器端返回结果解析—JSON 解析.....	425
12.2 本章案例实现流程	426
12.3 第三方 Jar 包导入.....	427
12.4 客户端与服务端交互工具类设计	430
12.5 “省控线查询”模块与服务端交互的实现.....	434
12.5.1 “省控线查询”模块与服务端交互流程	434
12.5.2 获取服务端数据	435
12.5.3 显示省控线数据列表.....	436
12.6 “历年录取线查询”模块与服务端交互的实现	438
12.6.1 “历年录取线查询”模块与服务端交互实现流程.....	438
12.6.2 获取服务端学校录取线和专业录取线.....	439
12.6.3 显示学校录取线和专业录取线列表.....	441
12.7 “报考咨询”模块与服务端交互的实现	443
12.7.1 “报考咨询”模块与服务端交互流程.....	443
12.7.2 获取历史问题列表.....	444
12.7.3 “提问”模块实现.....	446
12.7.4 “问题回复”模块实现.....	447
12.7.5 “查询问题”模块实现.....	450
12.8 课后练习	450
第 13 章 App 集成测试.....	451
13.1 单元测试	451
13.1.1 黑盒测试.....	451
13.1.2 白盒测试.....	453
13.2 集成测试	459
13.2.1 MySQL 数据库测试.....	459
13.2.2 Java Web 服务器访问 MySql 数据库集成测试	460
13.2.3 Android 客户端访问 Java Web 服务器集成测试.....	462
13.3 “豹考通”案例集成测试流程.....	464
13.3.1 集成测试流程	464
13.3.2 本书案例代码集成架构.....	480

13.4 课后练习	481
第 14 章 App 全栈工程师创业思维训练.....	483
14.1 App 全栈工程师创业思维训练的价值	483
14.1.1 有利于 App 全栈工程师自身的长远发展	483
14.1.2 有利于高校开展“互联网+”双创人才培养工作	484
14.2 App 全栈工程师创业思维训练基础.....	484
14.2.1 掌握定位理论，选好创业项目	484
14.2.2 了解创新者的窘境，思考创业项目的创新路径	485
14.2.3 理解精益创业，试探创业项目的商业模式	486
14.2.4 提升用户体验，推动创业项目成长.....	487
14.2.5 合理设置股权结构，保证初创公司长治久安和不断成长	488
14.3 “互联网+”创业思维训练实践.....	489
14.3.1 总体安排.....	489
14.3.2 “互联网+”创业思维实验安排	490
14.3.3 “互联网+”创业思维实验举例	492
14.4 课后练习	495

第 0 章 导读

在本章中，我们开门见山地迈入移动开发的大门，通过“豹考通”App 来介绍移动商务软件开发的知識模块与流程，并从功能和界面的角度阐述“豹考通”从设计到开发的全过程。在此基础上，我们从高校教师教学的形式出发，以项目案例驱动的形式辅助与教学，摒弃以往知识点讲解的教学形式，提高学习效果。

0.1 开发概述

App 是什么，有些读者会有些了解，简单地说 App 是对英文单词“Application”的简称，即智能手机的应用程序，无须连接 PC，直接运行在手机上的应用。App 开发对于企业来说，需要根据自身的具体发展情况和战略规划来权衡，一个好的企业，App 移动端开发在企业一定具有长期价值的一个重要组成部分。

App 拥有几个特点：

- (1) App 安装在移动设备上，使用方便，也便于营销。
- (2) App 融入了品牌元素，是品牌的一个具体的体现，用户可以享受到品牌的相关服务以及最新资讯。
- (3) App 与互联网相连，随时更新数据，提高用户体验。

0.2 App 开发流程

一个完整的软件开发分为网页端与移动端（App），而移动端又包括 Android 端和 IOS 端，使用的开发模式是移动应用开发常用的 MVC 开发模式。本书仅涉及 Android 端的开发。

- (1) 服务器框架搭建
- 开发流程如图 0-1 所示。

服务器端设计在一个移动项目设计中是很重要的部分，如果说客户端负责为用户展现数据，那么服务器端可以说是为用户准备数据的。只有服务器端将用户所需要的数据准备完毕后，客户端才有可能去展示。

- (2) 数据库设计与部署

对于移动 App 开发，数据库的选择就显得至关重要，因为数据存储结构、读取速度直接影响到用户体验，所以尽量使用轻量级数据库，这里我们使用的是 MySQL。

数据库开发分为逻辑结构设计和物理结构设计。逻辑结构设计是对数据库实体属性的确定，E-R 图确定以及数据库存储结构进行设计，需要确定项目中所用到的字段并确定数据类型。物理结构设计就是在 MySQL 中创建相应的数据库表，对数据进行录入存储。最后需要将数据库与我们的后台服务器相连接。

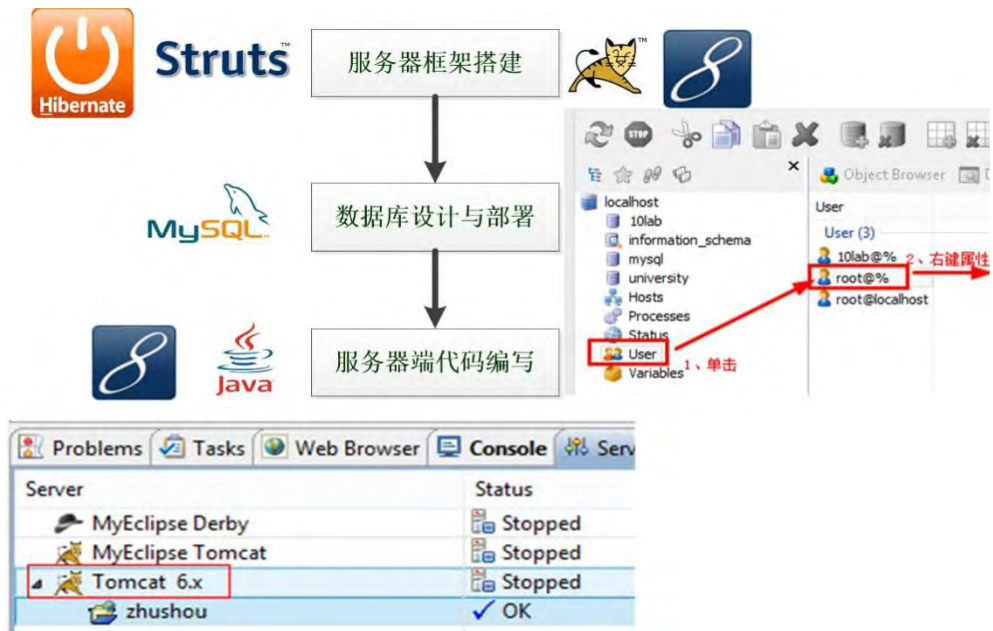


图 0-1 服务器端与数据库开发流程简图

(3) 服务器端实现

服务器端的实现是在MyEclipse8中实现，首先是要进行开发环境的配置。配置完成后，在MyEclipse8中创建“豹考通”工程。

(4) 客户端框架搭建

客户端开发流程简图如图 0-2 所示。



图 0-2 客户端开发流程简图

图 0-3、图 0-4、图 0-5 和图 0-6 为开发时的主要工作界面。

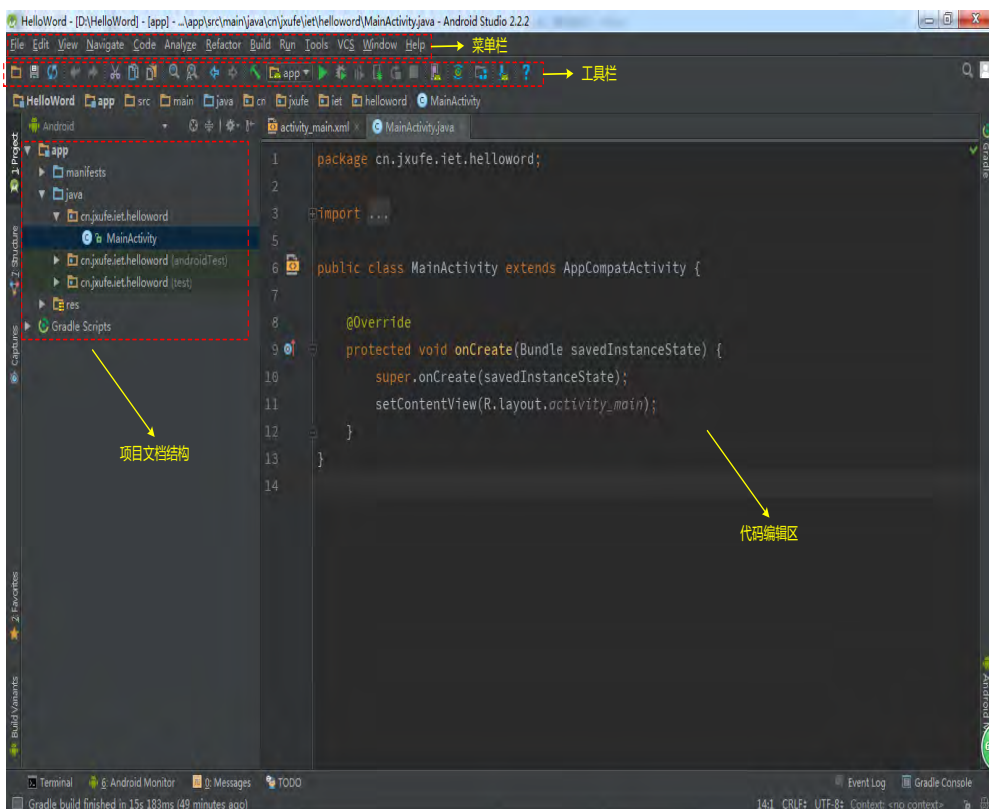


图0-3 用Android Studio开发客户端主界面

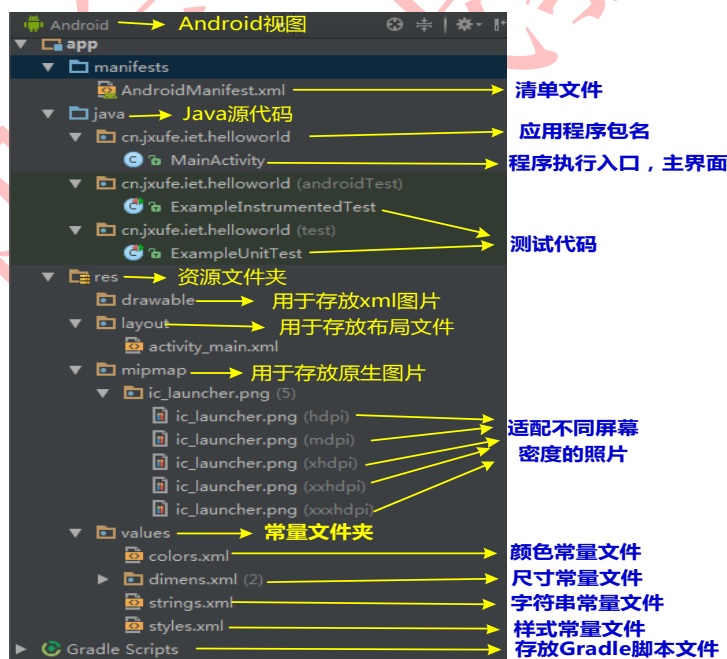


图 0-4 Android 视图中项目文档结构及其含义

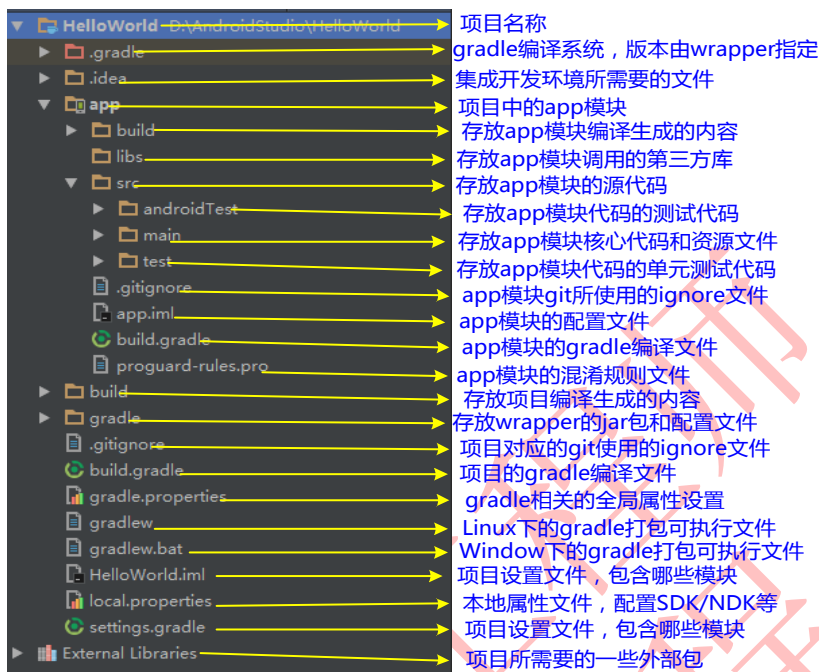


图 0-5 Project 视图中项目文档结构及其含义



图 0-6 客户端页面跳转

(5) 界面及素材设计和界面跳转设计

对于界面设计与模块类实现，可以分工同时进行。

项目素材由美工组制作完成，制作的标准需要按照 Android 开发图标要求制作，在开发前，项目负责人需要指定一个详细的设计文档，指出需要的素材以及规格。而界面与界面跳转设计，也需要通过讨论制定出想要的界面设计文档。其中需要注意的是，界面跳转不宜过多，除了必须跳转的界面之外，可以在当前界面中实现的功能尽量在当前页面实现，这也是影响用户体验的重要指标之一。

(6) 模块类的实现

根据项目的需求分析文档中列出的功能模块，在项目中分别创建模块文件夹，然后分别对功能模块进行设计与实现。

(7) 网络数据读取接口

网络数据读取接口由服务器端提供，包括网址和需要的参数，客户端发送相应

的参数请求从服务器端获取编译结果，最后由客户端呈现在用户眼前。网络数据接口形式如下所示。

<http://localhost/zhushou/RequestControlLinesAction.action?c=0&y=2013&s=14&b=0>。

（8）编译打包及测试

移动应用编译打包、测试及发布上线的简易流程如图 0-7 所示。

对于 Android 端应用，需要在项目中打包生成 APK 文件；编译打包完成后，可以在其他成员的手机手机上安装测试，看有没有问题出现。如果有，立即对程序进行调试；如果没有，就可以在商店中发布上线了。



图 0-7 移动应用编译打包、测试及发布上线流程简图

（9）发布上线

相对于 IOS 发布过程来说，Android 应用的发布比较简单；而 IOS 应用的发布审核相对来说更为严格，一些细节问题的出现都会让 Apple 公司拒绝，如果是发布在第三方商店，如 91 助手，审核相对顺利。

0.3 案例介绍

“豹考通”软件是一款面向全国高考学子和各大院校教师的软件。高考志愿填报对考生录取高校与专业,乃至未来的人生道路非常关键。由于信息不对称，考生填报志愿时比较盲目。研发的“豹考通”将是一款新型 App,服务考生和高校，借助智能手机帮助考生在填报志愿时掌握全面信息，提供数据跟踪记录和深度分析，供考生参考，同时在高校与考生之间搭建一条新的交流途径。

App 已经在各大 App 商店上架，读者可以先下载使用。以下提供 91 助手 Android 端下载地址。

<http://apk.91.com/Soft/Android/iet.jxufe.cn.android.score-3-1.2.html>

根据市场和往年高考情况综合考虑，“豹考通”实现了以下 7 个功能。

（1）省控线查询：用户可以查询各省市高考省控线信息，方便用户查找，不需要到网上去搜集信息。

(2) 投档线查询：用户可以查询全国各院校的投档线信息。

(3) 生成投档线曲线图：根据用户的信息选择，系统会为用户生成近几年的投档线和省控线的一个曲线图，更加直观地显示分数的变化波动情况，供用户参考。

(4) 预测投档线：根据往年投档线情况，采用系统的算法，根据用户不同的选择，预测今年各学校投档线情况。

(5) 推荐学校：推荐学校可以为用户推荐一些感兴趣并且有机会能投档的院校，帮助用户做出报考选择。

(6) 生成预测推荐报告：报告是根据用户选择的参数进行生成，报告的内容包括今年投档线的预测，和推荐的所有有机会投档大学等名单，为用户的最终选择提供参考。

(7) 关于我们：关于介绍参与开发“豹考通”软件的团队成员。

“豹考通”Android 客户端的主要界面见第 1 章体验部分。

案例中访问服务器的部分，均使用 www.xs360.cn 作为我们的服务器地址，所有的程序、软件包及源代码都可以从此网站中获取，当然读者也可以自己搭建本地服务器（访问局域网）。图 0-8 为客户端服务器网络连接示意图。

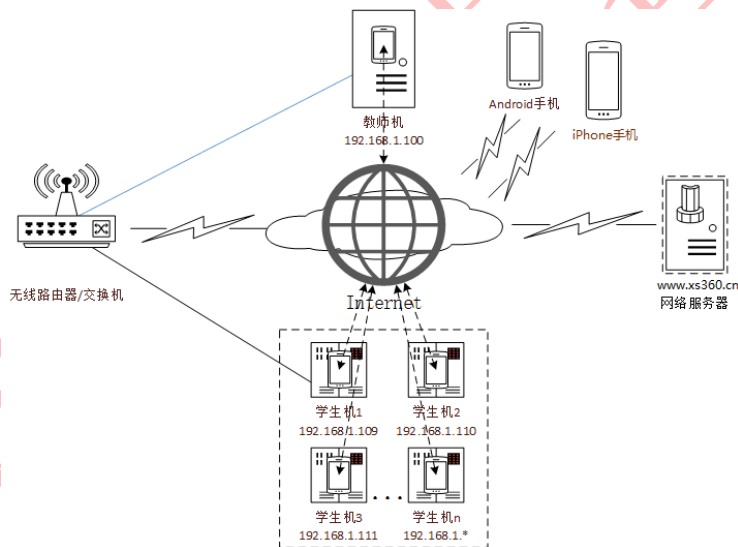


图 0-8 客户端服务器网络连接示意图

0.4 本书教学中所用到的学习案例代码

本书教学中所用到的案例代码，各个案例包参见第 13 章集成测试中§13.3.1 简介，详细的内容请参见网络下载的程序包。此处从略。

0.5 配置局域网运行和开发环境

客户端访问网络数据，既可以通过域名（www.xs360.cn）或域名指向的 IP 地址联网访问远程服务器获得数据，又可以搭建临时局域网服务器通过 IP 地址来访问本地服务器获取数据。本节将介绍如何搭建支持 App 客户端可访问的本地服务器和如何修改的客户端的请求 URL 链接。

由于“豹考通”服务器端代码是由 JSP 编写，如果需要将此服务器代码在本地机运行，则需用到的软件包括 Tomcat、JDK 和 Mysql。为了便于教学和测试，本书采用绿色运行环境 JTM-v1.1（一键搭建运行 JavaEE 应用的工具）为例，将“豹考通”Web 端部署在本地电脑并运行起来。

将实验室的教师端电脑与学生端电脑互联在一个局域网内，由教师或每组组长在电脑上将“豹考通”本地服务器运行起来（启动方法参见第 1 章），其他学生机进行调用，如图 0-9 网络拓扑结构所示，不同的小组既可以连接本组搭建的服务器，如小组 1 使用服务器 1，小组 2 使用服务器 2 等，也可以连接教师端的服务器，只需将请求的 URL 地址修改为教师端或任何一台学生机服务器的 IP 地址即可，如 192.168.1.100（教师机），192.168.1.101（小组 1 服务器）。完成上述 URL 地址修改后，将“豹考通”软件安装在学生机中 Android 模拟器上就可以访问本地搭建的服务器数据，如果安装到 Android 手机中测试，只需将 Android 手机连接到与服务器同一局域网内就可以成功获取请求（具体方法详见第一章）。

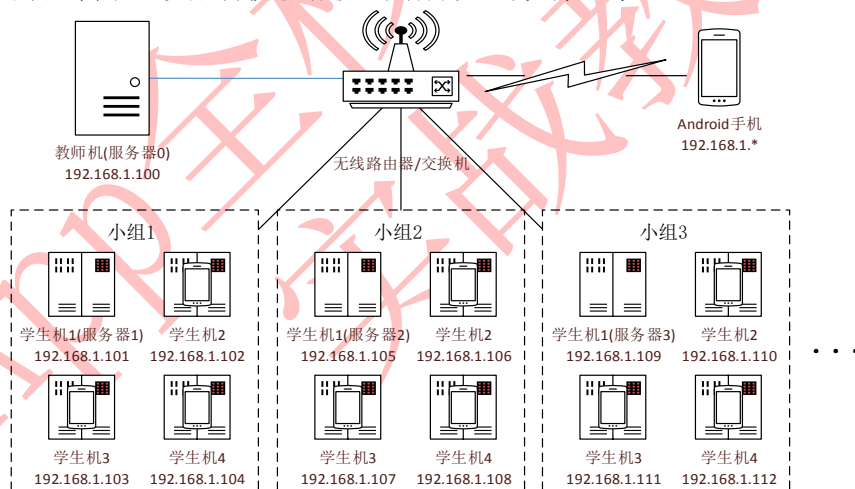
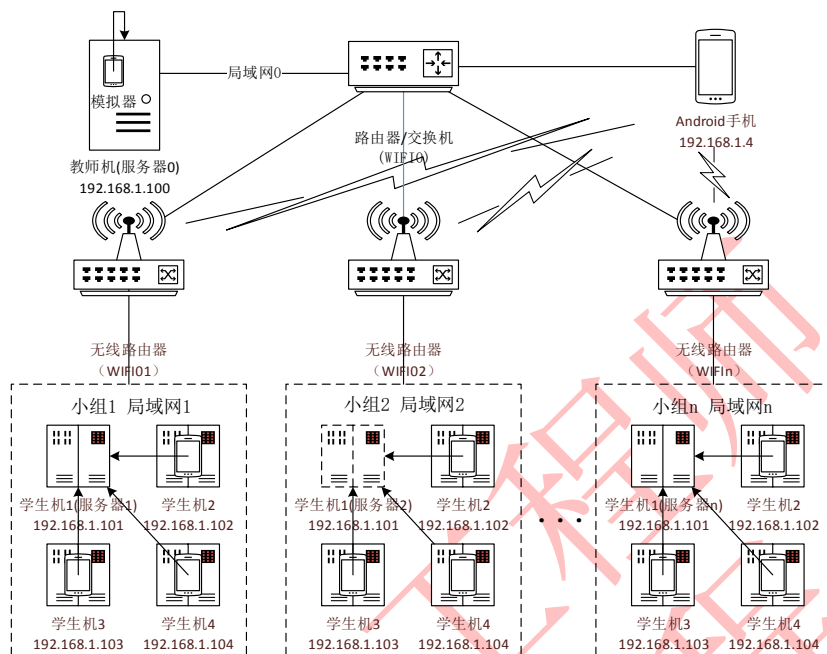


图 0-9 访问本地服务器网络连接拓扑图（一）

如果实验室的机房网络拓扑图如图 0-10 所示，由于各小组之间都被不同的无线路由器分割成不同的局域网，则整个访问规则服务器的只能是小组 1 使用服务器 1，小组 2 使用服务器 2，即本组只能访问本组局域网的服务器，而教师机的模拟器也只能访问本机的 IP 地址。

本地服务器的运行对交换机或路由器的设置上没有特殊要求，也不需要联网，只要按照默认配置使用。



0.6 本书内容简介

本书第 1 章 App 实战应用体验，运行 JVM 服务端和客户端，体验“豹考通”项目；第 2 章为项目需求分析；第 3 章体验一个包含 Android 客户端、服务端、数据库的开发微案例；第 4 和第 5 章把“豹考通”划分为四个模块，用本地临时数组分别进行实现；第 6 章采用 TabHost 多页面切换，将前述四个模块合并。第 7 章实现学校录取分数线趋势图的绘制；第 8 和第 9 章是本地 SQLite 数据库的管理与使用；第 10 章使用 MySQL 管理 App 的网络数据库；第 11 章开发网络服务端管理程序；第 12 章实现 Android 客户端与服务端的交互程序；第 13 章介绍 App 集成测试的基本方法；第 14 章为 App 全栈工程师创业思维训练。

0.7 课后练习

1. 移动 App 开发一般包括哪几个部分，并简要描述每个部分的流程。
2. 下载“豹考通”App，思考其界面跳转设计的思路，并提出自己的见解。
3. 搭建本地服务器，并在网页中试用 Web 版应用。

第4章 无对话框的单页面设计

通过第3章综合小案例的学习，我们学习了从服务端、客户端、数据库交互的过程。从本章开始我们把“豹考通”教学版分模块逐章实现，本章主要是用本地模拟数据代替获取服务端数据实现“历年省控线查询”和“历年录取线查询”两个模块，如图4-1和图4-2所示。

历年高考省控线查询

年份 生源地 批次 类别

2014 江西 不限 不限

江西地区2014年省控线

年份	类别	批次	省控线
2014	文史	一本	524
2014	理工	一本	526
2014	文史	二本	390
2014	理工	专科	230

图4-1 “历年省控线查询”界面

历年录取线查询

学校省份 选择学校 批次 类别 查询

江西 江西财经大学 一本 文史

江西农业大学一本文史类
在上海地区历年录取线

年份	省控线	投档线	平均分	最高分
2014	524	521	528	548
2013	532	522	532	552
2012	533	523	543	553
2011	511	501	521	541
2010	510	510	520	540

历年分数与省控线差值趋势图

图4-2 “历年录取线查询”界面

4.1 本章案例实现流程

在“省控线查询”模块中，先是实现模块整体界面布局，然后实现4个Spinner下拉列表，接下来对4个列表监听实现省控线标题变化，最后用ListView控件静态实现省控线列表，如下图4-3所示。



图4-3 “省控线查询”模块实现的页面流程

在“历年录取线查询”模块中，先是实现模块整体界面布局，然后对“历年录取线查询”筛选条件栏实现，接下来对筛选条件栏监听实现历年录取线查询标题变化，最后用 ListView 控件静态实现录取线列表和专业录取线，如图 4-4 所示。

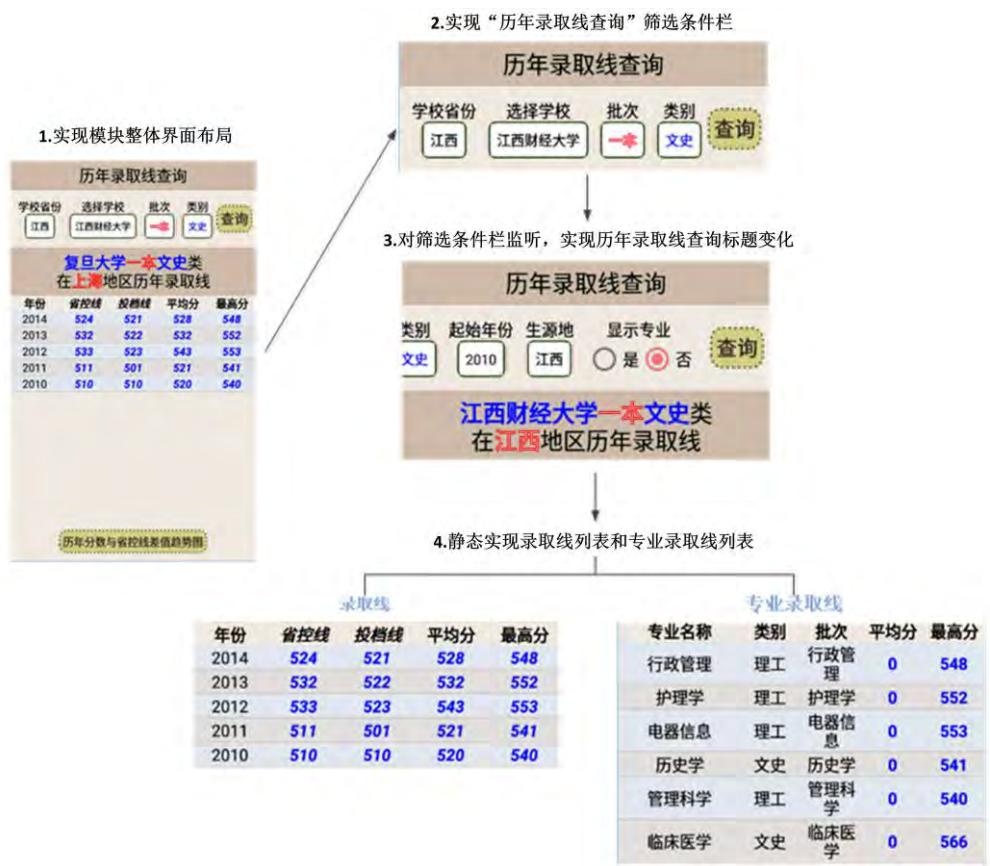


图4.4 “历年录取线查询”模块实现的界面流程

4.2 “历年省控线查询”模块实现

4.2.1 界面设计

“省控线查询”模块整体采用线性布局，布局内部结合 TextView(文本显示框)、Spinner(列表)、(HorizontalScrollView) 水平滚动条等组件组合实现，具体请参考图 4-5。



图4-5 “省控线查询”模块整体布局

打开 strings.xml 文件，在 resources 标签下定义资源变量，代码如下。

程序清单：Code0401\app\src\main\res\values\strings.xml

```
1 <resources>
2 <string name="batch">批次</string>
3 <string name="category">类别</string>
4 <string name="year">年份</string>
5 <string name="sourceArea">生源地</string>
6 <string name="schoolArea">学校省份</string>
7 <string name="controllLine">省控线</string>
8 <string name="controllLineTitle">历年高考省控线查询</string>
9 </resources>
```

Spinner 列表显示框因为有边框，所以需要自定义图片作为背景。在 drawable 文件夹下新建 border.xml 图片，代码如下。

程序清单：Code0401\app\src\main\res\drawable\border.xml

```
1 <shape xmlns:android="http://schemas.android.com/apk/res/android"
2     android:shape="rectangle">
3     <solid android:color="#00000000" /><!-- 无填充 -->
4     <corners android:radius="5dp"/>
5     <padding
```

```

6         android:bottom="2dp"
7         android:left="2dp"
8         android:right="2dp"
9         android:top="2dp" />
10    <stroke
11        android:width="1dp"
12        android:color="#004400" /><!-- 边框2dp, 颜色黑色 -->
13 </shape>

```

“省控线查询”模块整体布局代码如下。

程序清单：Code0401\app\src\main\res\layout\activity_main.xml

```

1    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2        xmlns:tools="http://schemas.android.com/tools"
3        android:layout_width="match_parent"
4        android:layout_height="match_parent"
5        android:background="#66ddccbb"
6        android:gravity="center_horizontal"
7        android:orientation="vertical">
8        <TextView
9            android:layout_width="match_parent"
10           android:layout_height="wrap_content"
11           android:background="#ccbbaa"
12           android:gravity="center"
13           android:padding="5dp"
14           android:text="@string/controlLineTitle"
15           android:textColor="#000000"
16           android:textSize="20sp" /><!-- 标题文本 -->
17        <HorizontalScrollView
18            android:layout_width="wrap_content"
19            android:layout_height="wrap_content"
20            android:layout_gravity="center_horizontal" >
21            <LinearLayout
22                android:layout_width="wrap_content"
23                android:layout_height="wrap_content"
24                android:gravity="center_horizontal"
25                android:orientation="horizontal"
26                android:padding="5dp" >
27                <LinearLayout
28                    android:layout_width="wrap_content"
29                    android:layout_height="wrap_content"
30                    android:gravity="center_horizontal"
31                    android:orientation="vertical"
32                    android:padding="5dp" >
33                    <TextView
34                        android:layout_width="wrap_content"
35                        android:layout_height="wrap_content"
36                        android:text="@string/year"
37                        android:textSize="16sp"

```



```
38         android:textColor="#000000" />
39     <Spinner
40         android:id="@+id/yearSpinner"
41         android:layout_width="45dp"
42         android:layout_height="30dp"
43         android:background="@drawable/border" />
44 </LinearLayout>
45 <LinearLayout
46     android:layout_width="wrap_content"
47     android:layout_height="wrap_content"
48     android:gravity="center_horizontal"
49     android:orientation="vertical"
50     android:padding="5dp" >
51     <TextView
52         android:layout_width="wrap_content"
53         android:layout_height="wrap_content"
54         android:text="@string/sourceArea"
55         android:textSize="16sp"
56         android:textColor="#000000" />
57     <Spinner
58         android:id="@+id/areaSpinner"
59         android:layout_width="40dp"
60         android:layout_height="30dp"
61         android:background="@drawable/border" />
62 </LinearLayout>
63 <LinearLayout
64     android:layout_width="wrap_content"
65     android:layout_height="wrap_content"
66     android:gravity="center_horizontal"
67     android:orientation="vertical"
68     android:padding="5dp" >
69     <TextView
70         android:layout_width="wrap_content"
71         android:layout_height="wrap_content"
72         android:text="@string/batch"
73         android:textSize="16sp"
74         android:textColor="#000000" />
75     <Spinner
76         android:id="@+id/batchSpinner"
77         android:layout_width="40dp"
78         android:layout_height="30dp"
79         android:background="@drawable/border" />
80 </LinearLayout>
81 <LinearLayout
82     android:layout_width="wrap_content"
83     android:layout_height="wrap_content"
84     android:gravity="center_horizontal"
85     android:orientation="vertical"
86     android:padding="5dp" >
```



```

87         <TextView
88             android:layout_width="wrap_content"
89             android:layout_height="wrap_content"
90             android:text="@string/category"
91             android:textSize="16sp"
92             android:textColor="#000000" />
93         <Spinner
94             android:id="@+id/categorySpinner"
95             android:layout_width="40dp"
96             android:layout_height="30dp"
97             android:background="@drawable/border" />
98     </LinearLayout>
99 </LinearLayout>
100 </HorizontalScrollView><!-- 选择条件 -->
101 <TextView
102     android:id="@+id/controlLineTitle"
103     android:layout_width="match_parent"
104     android:layout_height="wrap_content"
105     android:background="#ccbbaa"
106     android:gravity="center"
107     android:padding="10dp"
108     android:textColor="#000000"
109     android:textSize="20sp" /><!-- 结果标题 -->
110 <LinearLayout
111     android:id="@+id/resultTitle"
112     android:layout_width="match_parent"
113     android:layout_height="wrap_content"
114     android:orientation="horizontal" >
115     <TextView
116         android:layout_width="0dp"
117         android:layout_height="wrap_content"
118         android:layout_marginLeft="10dp"
119         android:layout_marginRight="10dp"
120         android:layout_weight="1"
121         android:gravity="center"
122         android:text="@string/year"
123         android:textColor="#000000"
124         android:textStyle="bold" />
125     <TextView
126         android:layout_width="0dp"
127         android:layout_height="wrap_content"
128         android:layout_marginRight="10dp"
129         android:layout_weight="1"
130         android:gravity="center"
131         android:text="@string/category"
132         android:textColor="#000000"
133         android:textStyle="bold" />
134 <TextView
135     android:layout_width="0dp"

```

```

136         android:layout_height="wrap_content"
137         android:layout_marginRight="10dp"
138         android:layout_weight="1"
139         android:gravity="center"
140         android:text="@string/batch"
141         android:textColor="#000000"
142         android:textStyle="bold" />
143     <TextView
144         android:layout_width="0dp"
145         android:layout_height="wrap_content"
146         android:layout_marginRight="10dp"
147         android:layout_weight="1"
148         android:gravity="center"
149         android:text="@string/controlline"
150         android:textColor="#000000"
151         android:textStyle="bold|italic" />
152 </LinearLayout>
153 <!-- 列表项标题 -->
154 <ListView
155     android:id="@+id/controllineList"
156     android:layout_width="match_parent"
157     android:layout_height="wrap_content" >
158 </ListView>
159 <!-- 结果列表 -->
160 </LinearLayout>

```

4.2.2 Spinner 实现下拉列表选项

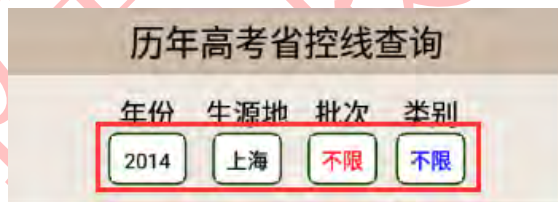


图4-6 省控线查询Spinner列表

4 个下拉列表采用 Spinner 控件实现，如图 4-6 所示，每个 Spinner 列表需要自己的布局、数据源和适配器 Adapter。在四个 Spinner 列表的布局中，使用了三种布局，其中年份和生源地的布局一样，因此创建三个布局文件即可。三种布局文件比较相似，只是颜色不一样，为了节省代码，使用 style 样式表进行实现。三个布局文件名分别是 spinner_item_batch.xml、spinner_item_category.xml 和 spinner_item_year.xml。

首先，在样式表 style.xml 文件中添加 spinner_itemStyle 样式，代码如下。

程序清单：Code0402\app\src\main\java\cn\jxufe\iet\code0402\ControlLineActivity.java

```
1  /*显示省控线列表 */
2  GetService getService=new GetService();
3  controllineList=  getService.setControllineList();//为 controllineList 集合
   添加集合内容
4  ControllineAdapter controllineAdapter=new ControllineAdapter
5      (this,controllineList);
6  controllineListView.setAdapter(controllineAdapter);
7  //关联 ListView 控件显示列表
```

4.3 “历年录取线查询” 模块实现

4.3.1 界面设计

“历年省控线查询”模块整体采用线性布局，布局内部结合 TextView（文本显示框）、Spinner（列表）、HorizontalScrollView（水平滚动条）、Button（按钮）等组件组合实现，具体请参考图 4-9。

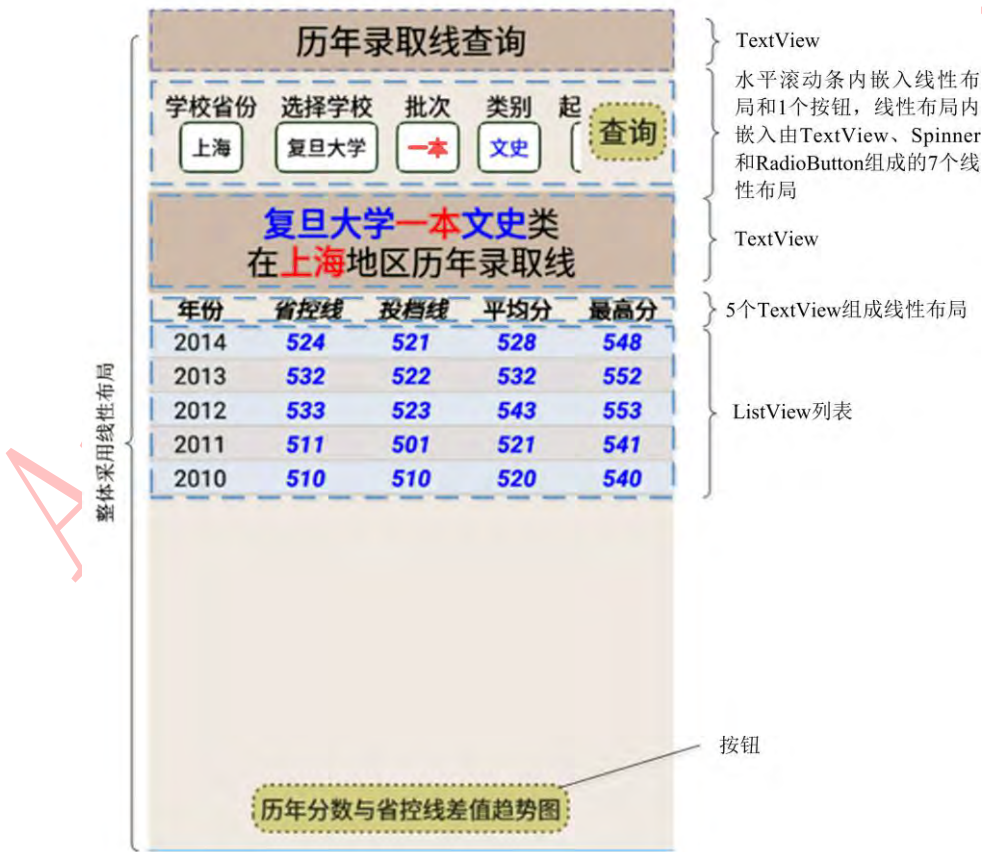


图 4-9 “省控线查询” 模块整体布局

打开 strings.xml 文件，在 resources 标签下定义资源变量，代码如下。

程序清单：Code0403\app\src\main\res\values\strings.xml

```

1  <resources>
2      <string name="chooseSchool">选择学校</string>
3      <string name="enrollScoreTitle">历年录取线查询</string>
4      <string name="showMajor">显示专业</string>
5      <string name="major">专业名称</string>
6      <string name="yes">是</string>
7      <string name="no">否</string>
8      <string name="startYear">起始年份</string>
9      <string name="enterLine">投档线</string>
10     <string name="enterLineRank">投档排名</string>
11     <string name="max_score">最高分</string>
12     <string name="average_score">平均分</string>
13     <string name="averageTrend">历年分数与省控线差值趋势图</string>
14     <string name="batch">批次</string>
15     <string name="category">类别</string>
16     <string name="year">年份</string>
17     <string name="sourceArea">生源地</string>
18     <string name="schoolArea">学校省份</string>
19     <string name="search">查询</string>
20 </resources>

```

Spinner 列表显示框因为有边框，所以需要自定义图片作为背景。在 drawable 文件夹下新建 border.xml 图片，代码如下。

程序清单：Code0403\app\src\main\res\drawable\border.xml

```

1  <shape xmlns:android="http://schemas.android.com/apk/res/android"
2      android:shape="rectangle">
3      <solid android:color="#00000000" /><!-- 无填充 -->
4      <corners android:radius="5dp"/>
5      <padding
6          android:bottom="2dp"
7          android:left="2dp"
8          android:right="2dp"
9          android:top="2dp" />
10     <stroke
11         android:width="1dp"
12         android:color="#004400" /><!-- 边框1dp，颜色黑色 -->
13 </shape>

```

查询按钮和历年分数与省控线差值按钮设置背景图片，因为按钮按下和抬起背景图片是两种状态，于是创建 title_btn.xml 背景图片，采用 selector 标签设置各种状态对应显示的图片，状态显示图片分别为 pressed.xml 和 unpressed.xml。

pressed.xml 自定义图片代码如下。

程序清单: Code0403\app\src\main\res\drawable\pressed.xml

```

1 <shape xmlns:android="http://schemas.android.com/apk/res/android"
  android:shape="rectangle" >
2   <corners android:radius="10dp"/>
3   <solid android:color="#ffff00"/><!-- 填充的颜色 -->
4   <stroke android:color="#666600"
5     android:width="2dp"/><!-- 边框的颜色 -->
6 </shape>

```

unpressed.xml 自定义图片代码如下。

程序清单: Code0403\app\src\main\res\drawable\unpressed.xml

```

1 <shape xmlns:android="http://schemas.android.com/apk/res/android"
  android:shape="rectangle" >
2   <corners android:radius="10dp"/>
3   <solid android:color="#66aaaa00"/><!-- 填充的颜色 -->
4   <stroke android:color="#666600"
5     android:width="1dp"
6     android:dashWidth="2dp"
7     android:dashGap="3dp"/><!-- 边框的颜色 -->
8 </shape>

```

title_btn.xml 自定义图片代码如下。

程序清单: Code0403\app\src\main\res\drawable\title_btn.xml

```

1 <selector xmlns:android="http://schemas.android.com/apk/res/android" >
2   <item android:state_pressed="true"
3     android:drawable="@drawable/pressed"></item>
4   <item android:state_selected="true"
5     android:drawable="@drawable/pressed"></item>
6   <item android:state_pressed="false"
7     android:drawable="@drawable/unpressed"></item>
8 </selector>

```

“历年录取线查询”模块整体布局代码如下。

程序清单: Code0403\app\src\main\res\layout\activity_main.xml

```

1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="match_parent"
4   android:layout_height="match_parent"
5   android:background="#66ddccbb"
6   android:orientation="vertical"
7   tools:context="cn.jxufe.iet.code0404.EnrollScoreActivity">
8   <TextView
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content"
11    android:background="#ccbbaa"
12    android:gravity="center"
13    android:padding="5dp"
14    android:text="@string/enrollScoreTitle"

```

4.3.3 事件监听器

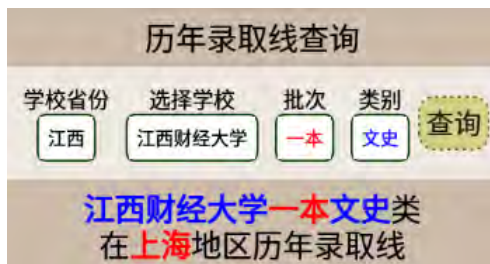


图4-11 “录取线查询”事件监听

历年录取线筛选条件栏实现 5 个 Spinner 列表完成后，点击查询按钮后，历年录取线标题栏标题还不能根据选择的信息进行切换，学校 Spinner 列表还没实现，如图 4-11 所示。学校列表在选择完学校所在省份后，更新学校列表，所以需要借助事件监听器进行实现。为了实现学校名称下拉列表随着选择学校省份变化和历年录取线标题栏灵活性变化，分三步进行实现：

(1) 创建 getSchool 方法，在 getSchool 方法中实现学校列表；

(2) 在主类 EnrollScoreActivity 中定义 MyItemSelectedListener 内部监听类对 6 个 Spinner 监听，同时点击学校省份后调用 getSchool 方法实现学校列表；

(3) 对 6 个 Spinner 列表进行注册。

第一步，在实现 getShool 方法前，需要实现 School 实例类，方便封装学校名称信息代码如下。

程序清单：Code0403\app\src\main\java\cn\jxufe\iet\code0403\bean\School.java

```

1  public class School implements Serializable { //学校封装类
2      private static final long serialVersionUID = 6952345113636426730L;
3      public static final String SCHOOL_NAME="schoolName";
4      public static final String SCHOOL_CODE="schoolCode";
5      public static final String SCHOOL_ID="schoolId";
6      private int schoolId;//学校 Id
7      private String schoolName;//学校名称
8      private int schoolCode;//学校编号
9      private boolean is211;//是否 211 院校
10     private boolean is985;//是否 985 院校
11     private boolean isMinistry;//是否直属院校
12     private String areaName;//学校所在地省份
13     public String getSchoolName() {
14         return schoolName;
15     }
16     public void setSchoolName(String schoolName) {
17         this.schoolName = schoolName;
18     }
19     public int getSchoolCode() {
20         return schoolCode;
21     }
22     public void setSchoolCode(int schoolCode) {

```

```

    合
8      }
9    }
10   ArrayAdapter<String> schoolAdapter = new ArrayAdapter<String>(
11       EnrollScoreActivity.this, R.layout.spinner_item_year, schools);
12   schoolSpinner.setAdapter(schoolAdapter);
13 }

```

第二步，定义 MyItemSelectedListener 内部监听类代码如下。

程序清单：Code0403\app\src\main\java\cn\jxufe\iet\code0403\EnrollScoreActivity.java

```

1   private class MyItemSelectedListener implements
   AdapterView.OnItemClickListener {
2       @Override
3       public void onItemClick(AdapterView<?> parent, View view,
4                               int position, long id) {
5           switch (parent.getId()) {
6               case R.id.schoolAreaSpinner:// 如果选择学校所在地
7                   schoolAreaId = position;
8                   getSchool(provinces.get(schoolAreaId));
9                   // 当省份变化时，对应显示的学校都会随之变化
10                  break;
11               case R.id.areaSpinner:// 如果是选择生源地
12                   sourceAreaId = position;
13                   break;
14               case R.id.categorySpinner:// 如果是选择文理科
15                   categoryId = position;// 序号从0开始
16                   break;
17               case R.id.batchSpinner:// 如果是选择批次
18                   batchId = position;
19                   break;
20               case R.id.schoolSpinner:// 如果是选择学校
21                   schoolId = position;// 获取学校名称
22                   break;
23               case R.id.yearSpinner:
24                   yearId = position;
25                   break;
26               default:
27                   break;
28           }
29       }
30       @Override
31       public void onNothingSelected(AdapterView<?> parent) {
32       }
33 }

```

第三步，定义完 MyItemSelectedListener 事件监听内部类后，需要对 6 个 Spinner 控件监听才能达到监听的效果，代码如下。

程序清单：Code0403\app\src\main\java\cn\jxufe\iet\code0403\EnrollScoreActivity.java

```

1  MyItemSelectedListener myItemSelectedListener = new
    MyItemSelectedListener();//实例化事件监听器
2  schoolAreaSpinner.setOnItemSelectedListener(myItemSelectedListener);
3  schoolSpinner.setOnItemSelectedListener(myItemSelectedListener);
4  categorySpinner.setOnItemSelectedListener(myItemSelectedListener);
5  batchSpinner.setOnItemSelectedListener(myItemSelectedListener);
6  yearSpinner.setOnItemSelectedListener(myItemSelectedListener);
7  sourceAreaSpinner.setOnItemSelectedListener(myItemSelectedListener);

```

实现完事件监听器对 6 个 Spinner 监听记录选择的信息，但是点击“查询”按钮还是不能实现录取线标题栏信息改变，这时需要对“查询”按钮采用匿名内部类监听方法对实现。对“查询”按钮监听，根据检查单选按钮的选择情况显示信息，具体代码如下。

程序清单：Code0403\app\src\main\java\cn\jxufe\iet\code0403\EnrollScoreActivity.java

```

1  searchBtn.setOnClickListener(new View.OnClickListener() {
2      @Override
3      public void onClick(View v) {
4          if (isSearchByMajor.isChecked()) { // 查询学校各专业的录取线
5              showMajorRecruitsResult();
6          } else { // 查询学校的录取线
7              showSchoolRecruits();
8          }
9      }
10 });

```

根据上面代码，我们可以发现 showMajorRecruitsResult 方法和 showSchoolRecruits 方法没有实现，showMajorRecruitsResult 方法为显示选择录取线结果，showSchoolRecruits 显示学校录取线结果，具体如何显示代码如下。

showMajorRecruitsResult 方法代码。

程序清单：Code0403\app\src\main\java\cn\jxufe\iet\code0403\EnrollScoreActivity.java

```

1  public void showMajorRecruitsResult() { // 显示专业的录取线信息
2      trendBtn.setVisibility(View.GONE); // 生成趋势图按钮不可见
3      schoolRecruitResultLayout.setVisibility(View.GONE); // 隐藏学校录取线信息标题
4      majorRecruitResultLayout.setVisibility(View.VISIBLE); // 显示专业录取线信息标题
5      titleView.setText(Html.fromHtml("<font color=blue><b>" +
6          schools.get(schoolId)
7          + "</b></font><font color=red><b>" + years[yearId]
8          + "</b></font>年在<br/><font color=blue><b>"
9          + provinces.get(sourceAreaId) + "</b></font>地区各专业录取线"));

```

showSchoolRecruits 方法代码。

程序清单：Code0403\app\src\main\java\cn\jxufe\iet\code0403\EnrollScoreActivity.java

```

1  public void showSchoolRecruits() { // 显示学校的录取线信息
2      titleView.setText(Html.fromHtml("<font color=blue><b>" +
3          schools.get(schoolId)
4              + "</b></font><font color=red><b>"
5              + batches[batchId]
6              + "</b></font><font color=blue><b>"
7              + categories[categoryId]
8              + "</b></font>类<br/>在<font color=red><b>"
9              + provinces.get(sourceAreaId) + "</b></font>地区历年录取线")); //
10     标题内容
11     schoolRecruitResultLayout.setVisibility(View.VISIBLE); // 显示学校录取线
12     各项标题
13     majorRecruitResultLayout.setVisibility(View.GONE); // 专业录取信息标题隐
14     藏
15     trendBtn.setVisibility(View.VISIBLE); // 生成趋势图按钮可见
16 }

```

4.3.4 ListView 列表

年份	省控线	投档线	平均分	最高分
2014	524	521	528	548
2013	532	522	532	552
2012	533	523	543	553
2011	511	501	521	541
2010	510	510	520	540

图4-12 学校录取线列表

专业名称	类别	批次	平均分	最高分
行政管理	理工	行政管理	528	548
护理学	理工	护理学	532	552
电器信息	理工	电器信息	543	553
历史学	文史	历史学	521	541
管理科学	理工	管理科学	520	540
临床医学	文史	临床医学	533	566

图4-13 专业分数线列表

录取线列表包含两部分：录取线列表和专业录取线列表，如图 4-12 和图 4-13 所示。实现 ListView 列表需要四步进行完成：(1)分别实现学校录取线列表布局和专业分数线列表；(2)使用临时集合数据源代替从服务端获取的数据；(3)自定义每个列表的适配器类，并实现其内部方法；(4)将适配器和列表进行关联。

第一步，实现学校录取线列表布局，在 Layout 文件夹下新建 list_item_school_recruit.xml 列表布局代码如下。

程序清单：Code0404\app\src\main\java\cn\jxufe\iet\code0404\EnrollScoreActivity.java

```
1  GetService getService=new GetService();
2  schoolRecruits=getService.setSchoolRecruits();
3  School_recruitAdapter school_recruitAdapter=new
   School_recruitAdapter(EnrollScoreActivity.this,schoolRecruits);
4  enrollResultView.setAdapter(school_recruitAdapter);
```

在 showMajorRecruitsResult 方法中实例化 GetService 对象获取省控线数据集合，并实例化 Major_recruitAdapter 对象，传入数据源，关联 ListView 控件显示列表，具体代码如下。

程序清单：Code0404\app\src\main\java\cn\jxufe\iet\code0404\EnrollScoreActivity.java

```
1  GetService getService=new GetService();//实例化从服务端获取数据类
2  majorRecruits=getService.setMajorRecruits();//获取专业信息数据
3  Major_recruitAdapter major_recruitAdapter=new
   Major_recruitAdapter(EnrollScoreActivity.this,majorRecruits);
4  enrollResultView.setAdapter(major_recruitAdapter);
```

4.4 课后练习

1. Spinner 列表尝试另外一种实现方式，在 XML 文件中通过 <string-array> 进行指定，然后为 Spinner 控件指定 android: entries 属性。
2. 在“历年录取线查询页面”的基础上添加 2015 年的数据，并将它们显示出来。
3. 将“历年录取线查询页面”中的 ListView 的奇数行的背景颜色改为黑色，字体颜色改为白色。
4. 请设计并实现如图 4-14 所示界面效果。

界面要求：

- a. 最上面为一张图片控件，采用提供图片资源显示。中间三行采用表格布局进行显示，其中 TextView 字体采用 22sp，第一行我 Spinner 控件，第二行和第三行第二类为 RadioButton。最下面为按钮，背景颜色为 #8080C0，宽度 200dp，高度 40dp。
- b. Spinner 控件设置背景编辑框，使用 shape 标签自定义图片进行实现。
- c. 使用省份数组显示 Spinner 列表，如图 4-15 (b) 所示。

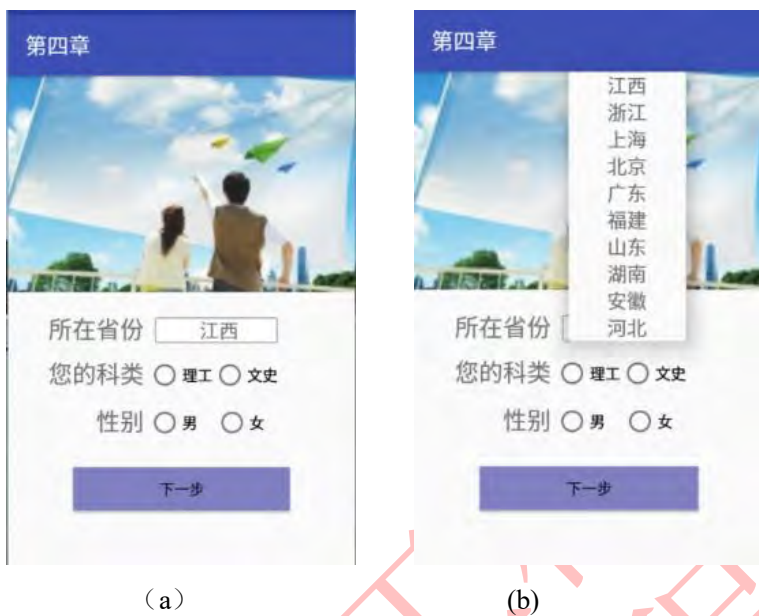


图4-14 习题4效果图

第5章 含对话框单页面界面设计

我们把“豹考通”教学版分模块逐章实现过程中，在第4章中已经用本地模拟数据代替获取服务端数据的实现“历年省控线查询”和“历年录取线查询”两个模块，本章主要用本地模拟数据临时代替服务端数据实现“报考咨询”模块，“个人基本信息”模块没有与服务端进行数据交互，本地实现即可，如图5-1和图5-2所示。



图5-1 “报考咨询”界面

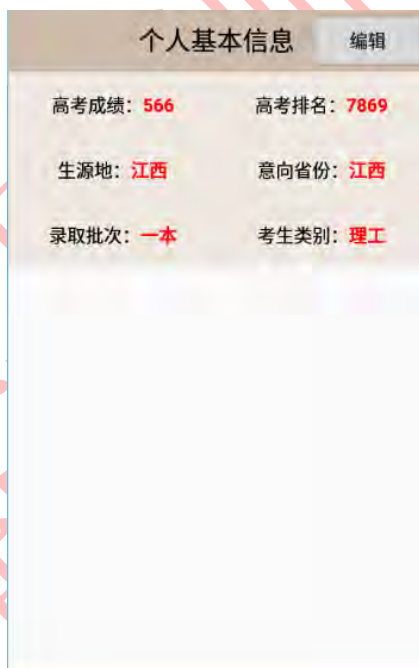


图5-2 “个人基本信息”界面

5.1 本章案例实现流程

在“报考咨询”模块中，先是实现模块整体界面布局，然后实现问题列表，对问题列表长按进行事件监听，长按能够弹出问题回复对话框，如图5-3所示。



图5-3 “报考咨询”模块实现流程图

在“个人基本信息”模块中，先是实现模块整体界面布局，然后通过 SharedPreferences 实现个人信息存储，接下来实现“个人信息填写对话框”，如下图 5-4 所示。



图5-4 “历年录取线查询”模块实现流程图

5.2 “报考咨询”模块实现

5.2.1 界面设计

“报考咨询”模块整体采用线性布局，布局内部结合 TextView（文本显示框）、EditText（编辑框）、Button（按钮）、ListView（列表）等组件组合实现，具体请参考图 5-5。



图5-5 “省控线查询”模块整体布局

打开 strings.xml 文件，在 resources 标签下定义资源变量，代码如下。

程序清单：Code0501\app\src\main\res\values\strings.xml

```

1  <resources>
2  <string name="app_name">Code0501</string>
3  <string name="consult">报考咨询</string>
4  <string name="submitQuestion">提问</string>
5  <string name="inputHint">请输入你想咨询的问题</string>
6  <string name="keywordHint">请输入搜索关键字</string>
7  <string name="save">保存</string>
8  <string name="selfInfoTitle">个人基本信息</string>
9  <string name="score">高考成绩</string>
10 <string name="inputScore">输入高考成绩</string>
11 <string name="rank">高考排名</string>
12 <string name="targetArea">意向省份</string>
13 <string name="recruitBatch">录取批次</string>
14 <string name="studentCategory">考试类别</string>
15 <string name="search">查询</string>
16 </resources>

```

5.4 课后练习

1. 为“个人基本信息界面”的编辑按钮添加样式，使之与“报考咨询界面”的按钮一致。

2. 尝试编写一个登录的小案例，该页面应该有一个登录按钮和两个输入框。当输入了账号 admin，密码 123 时即可弹出一个 Toast 提示“登录成功”，若账号或密码输入错误则 Toast 显示“登录失败”。登陆成功后需要将账号和密码存储在 SharedPreferences 中，在用户第二次打开案例时，程序会从 SharedPreferences 中取出账号和密码并将它们自动填写在输入框中。

3. 请设计并实现如图 5-15 所示界面效果。

界面要求：

- a) 界面中包含两个控件，文本显示框 **TextView** 和列表 **ListView**。
TextView 用于显示标题信息，标题内容为：**南昌景点介绍**，文字大小为：**24sp**，背景颜色为：**#ecbbaa**，对齐方式为**居中**，边距为 **10dp**。
- b) **ListView** 显示所有景点信息，一项代表一个景点，每个景点包含三部分信息，**景点图片**，**景点名称**，**景点简介**。**ListView** 的背景颜色为 **#aabbcc**，项与项之间的分割线大小为 **2dp**，颜色为**灰色** (**#aaaaaa**)。
- c) 在 **ListView** 的每一项中，包含三个控件：一个 **ImageView** 用于显示景点图片，两个 **TextView** 分别显示景点名称、景点简介。其中 **ImageView** 大小为：**100*75**，景点名称文字大小为：**20sp**，景点简介文字大小为：**12sp**，颜色为：**#0000ee**，**单行显示**，当内容超过宽度时，**省略后面的文字**，以点代替。

功能要求：

长按列表中的某一项，弹出如图所示删除提示对话框，如果用户选择**确定**，则**删除该项**信息，并及时**更新列表**；如果用户选择**取消**，则**不做删除操作**。

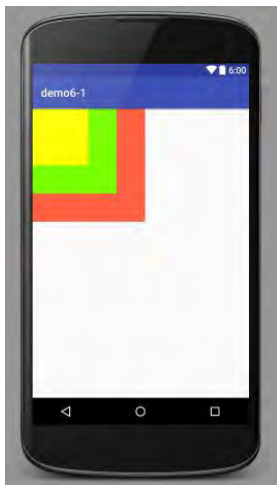


图6-14 FrameLayout布局效果演示

在“豹考通”教学版中，各个页面的切换实质是让选项卡中所选中的页面覆盖当前页面。

此时，我们发现在 `activity_main.xml` 的视图界面显示如下：



图6-15 activity_main视图界面

注意：如果要使用 TabHost，必须在布局文件中给 TabWidget 定义名为 `tabs` 的 id。除此之外，还需要给布局中的一个 FrameLayout 定义一个名为 `tabcontent` 的 id。这两个 id 是 android 系统自带的，如果不为这两个元素添加系统自带的 id，选项卡就无法正常使用。

中通常用来存放一些写好的控件特效。在“豹考通”教学版的选项卡模块中，在 `drawable` 文件夹里使用了 3 个 `xml` 文件来编写选项卡的特效，他们分别是 `tab.xml` (和 `layout` 文件夹下的布局文件是重名的，事实上这个名称可以随意设置)、`tab_selected.xml`、`tab_unselected.xml`。

其中，`tab_selected.xml` 表示的是当选项卡被选中时会呈现出的样式，`tab_unselected.xml` 表示的是当选项卡未被选中时呈现的样式。`tab.xml` 文件整合了前两个文件，用于设置当选项卡处于不同状态下应该加载哪种布局文件。

我们只需要在 `res-drawable` 文件夹中分别建立以下三个布局文件即可：

选项卡状态设置文件：Code0601\app\src\main\res\drawable\tab.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <!--selector 标签经常用来操作控件的背景-->
3 <selector xmlns:android="http://schemas.android.com/apk/res/android">
4 <!--当控件被按下时，将控件样式设置为 tab_selected.xml-->
5 <item android:state_pressed="true"
6     android:drawable="@drawable/tab_selected"></item>
7 <!--当控件被选中时，将控件样式设置为 tab_selected.xml-->
8 <item android:state_selected="true"
9     android:drawable="@drawable/tab_selected"></item>
10 <!--当控件未被按下时，将控件样式设置为 tab_unselected.xml-->
11 <item android:state_pressed="false"
12     android:drawable="@drawable/tab_unselected"></item>
13 <!--当控件未被选中时，将控件样式设置为 tab_unselected.xml-->
14 <item android:state_selected="false"
15     android:drawable="@drawable/tab_unselected"></item>
16 </selector>
```

选项卡样式：Code0601\app\src\main\res\drawable\tab_selected.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android"
3     android:shape="rectangle"><!--设置控件为矩形-->
4     <corners android:radius="5dp"/><!--设置控件的圆角弧度-->
5     <solid android:color="#aaaaaa"/><!-- 填充的颜色 -->
6 </shape>
```

选项卡未被选中/按下时的样式：Code0601\app\src\main\res\drawable\tab_unselected.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android"
3     android:shape="rectangle">
4     <corners android:radius="5dp"/>
5     <solid android:color="#000000"/><!-- 填充的颜色 -->
6     <gradient <!--用于设置颜色的过渡-->
```

```
7         android:startColor="#aa444444"
8         android:centerColor="#aa666666"
9         android:endColor="#aa444444"
10        android:type="linear"/>
11    </shape>
```

当我们将这几个 xml 文件编写完后,可以发现在 layout/tab.xml 布局文件中已经不再报错了。

6.3.4 初始化选项卡设计

我们已经在布局文件中完成了各个控件的插入,但此时还并不能看到选项卡的效果,因为我们并没有为选项卡加载内容,接下来我们进入 MainActivity.java 文件,来正式为选项卡设置内容。

首先,我们需要为每个选项卡设置标签 tag 和标题 title。tag 指代的是选项卡每个子项的名字, title 指代的是选项卡每个子项的内容,在“豹考通”教学版中的选项卡有四个选项,那么我们就分别为 tag 和 title 建立一个数组来存储信息,同时我们要初始化选项卡控件。代码如下:

页面主程序java文件:Code0601\app\src\main\java\cn\jxufe\iet\MainActivity.java

```
1
2 public class MainActivity extends Activity {
3     <!--初始化选项卡控件-->
4     private TabHost mTabHost;
5     <!--选项卡的标签-->
6     private String[] tags=new String[]{"page1","page2","page3","page4"};
7     <!--选项卡的内容-->
8     private String[] titles=new String[]{"第一页","第二页","第三页","第四页"};
9 }
10
```

上述基本准备工作完成后,我们需要了解一个新的概念—Fragment。Fragment 是 Android3.0 引入的新 API,译为“片段”,我们可以将它理解为 Activity 中的片段或者子模块。Fragment 必须被嵌入到 Activity 中使用,在同一个 Activity 中可以包含多个 Fragment。

在“豹考通”教学版中,我们便使用到了 TabHost+Fragment 的技术,用户通过单击选项卡完成页面的切换,在切换过程中整个 Activity 并没有发生跳转,实际上发生变化的仅仅是 Activity 中的 Fragment 部分。如下图所示:

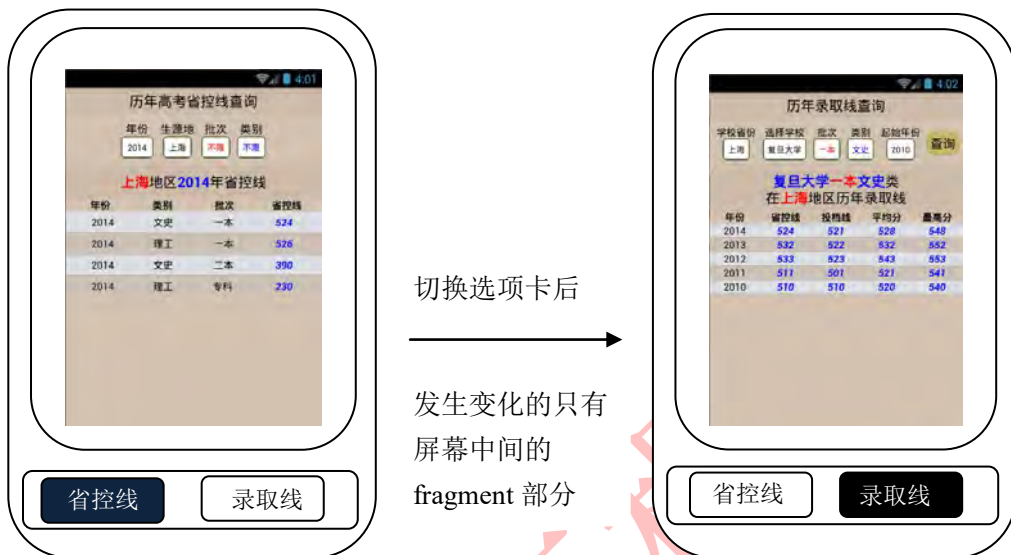


图6-17 选项卡效果展示

如果希望使用 `fragment` 来完成不同选项卡之间内容的跳转，我们需要首先让 `MainActivity` 集成 `FragmentManager`，并且声明 `onCreate()` 方法，代码如下：

页面主程序java文件:Code0601\app\src\main\java\cn\jxufe\iet\MainActivity.java

```

1 public class MainActivity extends FragmentActivity {
2     <!--初始化选项卡控件-->
3     private TabHost mTabHost;
4     <!--选项卡的标记-->
5     private String[] tags=new String[]{"page1","page2","page3","page4"};
6     <!--选项卡的内容-->
7     private String[] titles=new String[]{"第一页","第二页","第三页","第四页"};
8     <!--onCreate()方法-->
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11     }
12 }
```

此处要继承
FragmentActivity

接下来，我们为每个选项卡中填充内容，代码如下：

页面主程序java文件:Code0601\app\src\main\java\cn\jxufe\iet\MainActivity.java

```

1 public class MainActivity extends FragmentActivity {
2     private TabHost mTabHost;
3     private String[] tags=new String[]{"page1","page2","page3","page4"};
4     private String[] titles=new String[]{"第一页","第二页","第三页","第四页"};
5     protected void onCreate(Bundle savedInstanceState) {
6         super.onCreate(savedInstanceState);
7     }
8 }
```

第 7 章 绘制趋势图

7.1 本章案例流程



图7-1 引用自定义View
(Code0701)



图7-2 绘制坐标轴
(Code0702)

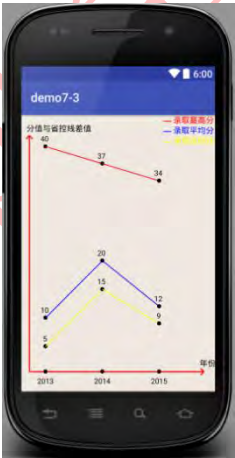


图7-3 根据模拟数据绘制趋势图
(Code0703)

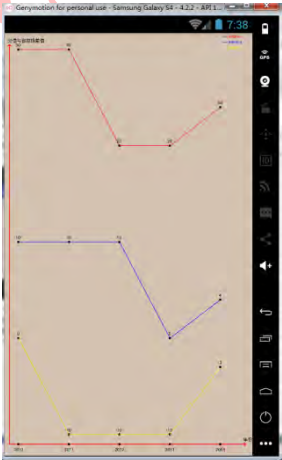


图7-4 根据录取线页面数据绘制趋势图
(Code0704)

7.2 绘图功能了解

为了让用户在填报院校, 评估院校时有更加清晰的思路, “豹考通” App 特地提供了一项分数线趋势图的功能, 趋势图模块可以将院校历年的省控线和录取线的差值展示在坐标轴上并连线, 以便让用户清晰地了解到近几年分数线的变动情况, 用户可以根据图形呈上升趋势或者下降趋势来判断自己填报的合理性。趋势图查询功能如下图所示。



图7-5 查询完录取线后, 点击
“历年分数与省控线差值趋势图”按钮

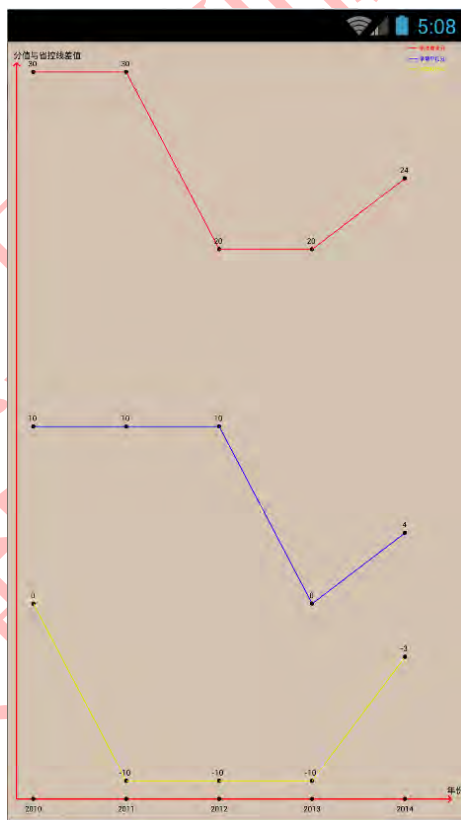


图7-6 进入趋势图页面

在本章提供的源码中, Code0701 对应第 7.4 节教学内容, Code0702 对应第 7.5 节的教学内容, Code0703 对应第 7.6 节的教学内容, Code0704 对应第 7.7 节的教学内容。

第 8 章 使用工具管理 SQLite 数据库

通过前面几章的学习，我们知道如何从数组中获取数据并将数据显示到相应的控件中去。从本章开始我们将引入本地数据库 SQLite，将学校表 School 和地区表 Area 中的数据保存在 SQLite 中，然后从 SQLite 取值并显示到控件中。本章主要是告诉大家如何使用工具管理 SQLite 数据库，其创建的数据库表如图 8-1 所示，再将准备好的学校数据和地区数据导入到 SQLite 数据中，具体数据如表 8-1 和表 8-2 所示。

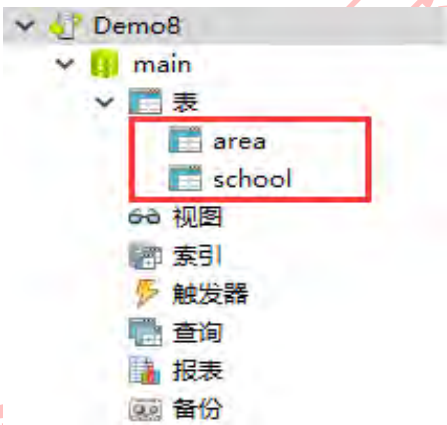


图 8-1 数据库表

表8-1 地区表(area)全部数据

areald（省市编码）	areaName（省市名称）
3	上海
10	浙江
13	江西
34	北京

表8-2 学校表(school)部分数据

schoolId	schoolName	schoolCode	areald	is985	is211	isMinistry
5	北京大学	10001	34	1	1	1
6	清华大学	10003	34	1	1	1
8	浙江大学	10335	10	1	1	1
14	复旦大学	10246	3	1	1	1
16	上海中医药大学	10268	3	0	0	0
22	同济大学	10247	3	1	1	1
27	中国人民大学	10002	34	1	1	1
28	北京理工大学	10007	34	1	1	0
31	上海大学	10280	3	0	1	0
39	中央财经大学	10034	34	0	1	1

8.1 了解 SQLite

SQLite 是一款轻量级、跨平台的关系型数据库。既然号称关系型数据库，那它就支持多数 SQL92 标准（比如视图、触发器、事务等），这里我们就不一一细说了。它的运算速度非常快，并且占用资源很少，通常只需要几百 KB 的内存就可以了，因而特别适合在移动设备上使用。Android 系统是内置了数据库的，SQLite 是 D. Richard Hipp 用 C 语言编写的开源嵌入式数据库引擎。它支持大多数的 SQL92 标准，并且可以在所有主要的操作系统上运行。

SQLite 由以下几个部分组成：SQL 编译器、内核、后端以及附件。SQLite 通过利用虚拟机和虚拟数据库引擎 (VDBE)，使调试、修改和扩展 SQLite 的内核变得更加方便。所有 SQL 语句都被编译成易读的、可以在 SQLite 虚拟机中执行的程序集。SQLite 支持 NULL、INTEGER、REAL、TEXT 和 BLOB 数据类型，分别代表空值、整型值、浮点值、字符串文本、二进制对象。SQLite 不仅支持标准的 SQL 语法，还遵循了数据库的 ACID 事务，所以只要你以前使用过其他的关系型数据库，就可以很快地上手 SQLite。而 SQLite 又比一般的数据库要简单地多，它甚至不用设置用户名和密码就可以使用。Android 正是把这个功能极为强大的数据库嵌入到了系统当中，使得本地持久化的功能有了一次质的飞跃。SQLite 系统架构如图 8-2 所示。

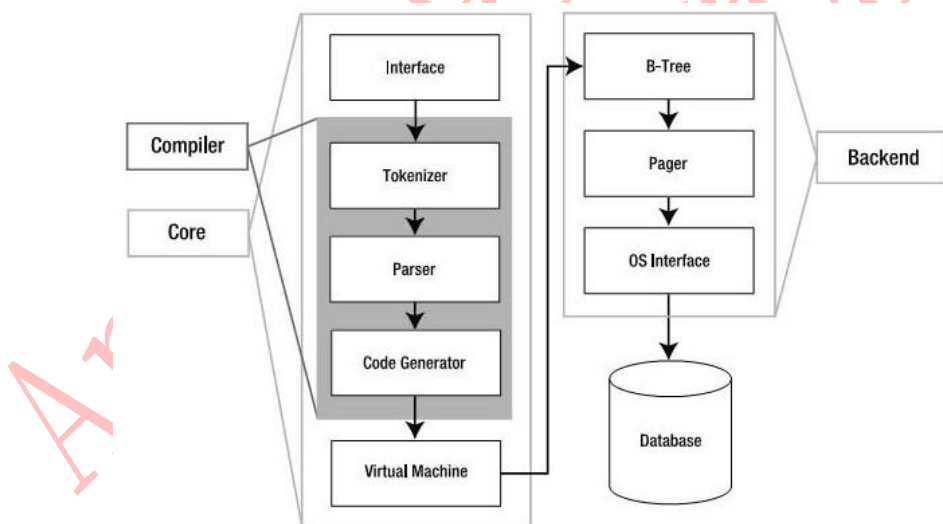


图 8-2 SQLite 系统架构

8.2 SQLite 优点

轻量级：先说它的第一个特色：轻量级。想必 SQLite 的作者很看重这个特性，连它的 Logo 都是用的“羽毛”，来显摆它的轻飘飘。SQLite 和 C/S 模式的数据库

第 9 章 SQLite 数据库的操作模块设计

9.1 本章案例流程

上章我们讲到如何使用工具（Navicat for SQLite）管理 SQLite 数据库，并创建好了本地数据库“bkt.sqlite”。本章将讲解如何使用代码操作 SQLite 数据库，同时将上一章创建好的数据库导入到项目中并成功运行。

使用代码操作 SQLite 数据库有两种方式：

- (1) 直接用代码创建数据库及表，并向表里添加数据，或是执行其他的 SQL 操作。
- (2) 将外部 SQLite 数据库导入到项目中，然后使用代码读取该 SQLite 数据库，并执行相应的 SQL 操作。



图9-1 Code0901 代码创建SQLite数据库



图9-2 引入外部SQLite数据库

9.2 用 SQL 代码创建数据库及表

9.2.1 项目结构

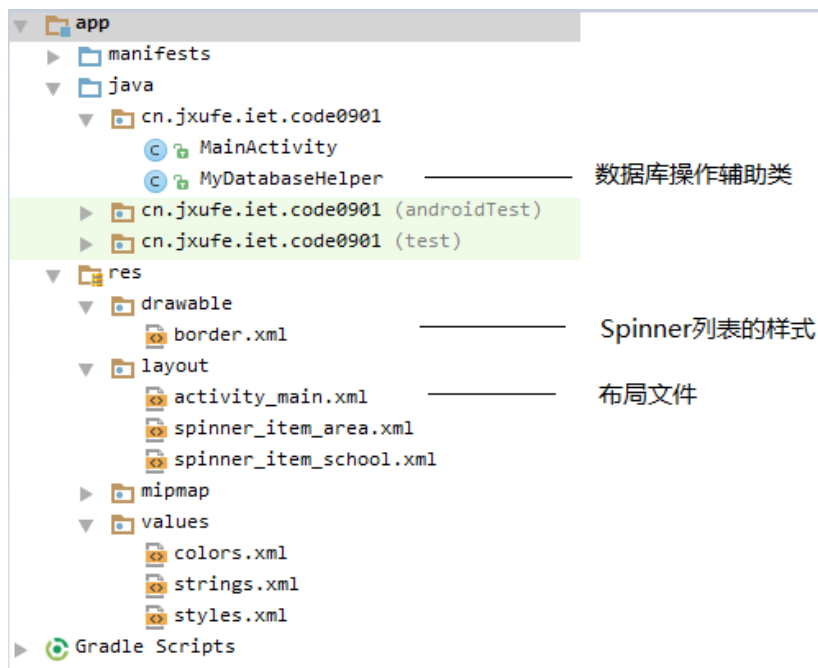


图9-3 Code0901项目结构

9.2.2 案例实现

首先我们直接通过一个小案例来看下第一种方式该怎么去实现对 SQLite 的操作，有关 SQLite 的属性和一些具体的用法稍后便会讲到。这里我们要在 Android Studio 中使用代码创建一个名为“bkt_local.db”的本地数据库，然后在这个数据库中新建一张地区表 area 和学校表 school。建表语句不会陌生吧？还是用基本的 SQL 语句创建，area 表的建表语句如下所示：

```
create table area(areaId integer primary key,areaName text);
```

school 表的建表语句如下所示：

```
1 create table school(
2     schoolId integer primary key,
3     schoolName text,
4     schoolCode integer,
5     areaId integer,
6     is985 integer,
7     is211 integer,
8     isMinistry integer);
```

然后我们怎么把以上建表语句整合在代码中并执行呢？我们先新建一个类

“java class”，如图 9-4 所示。

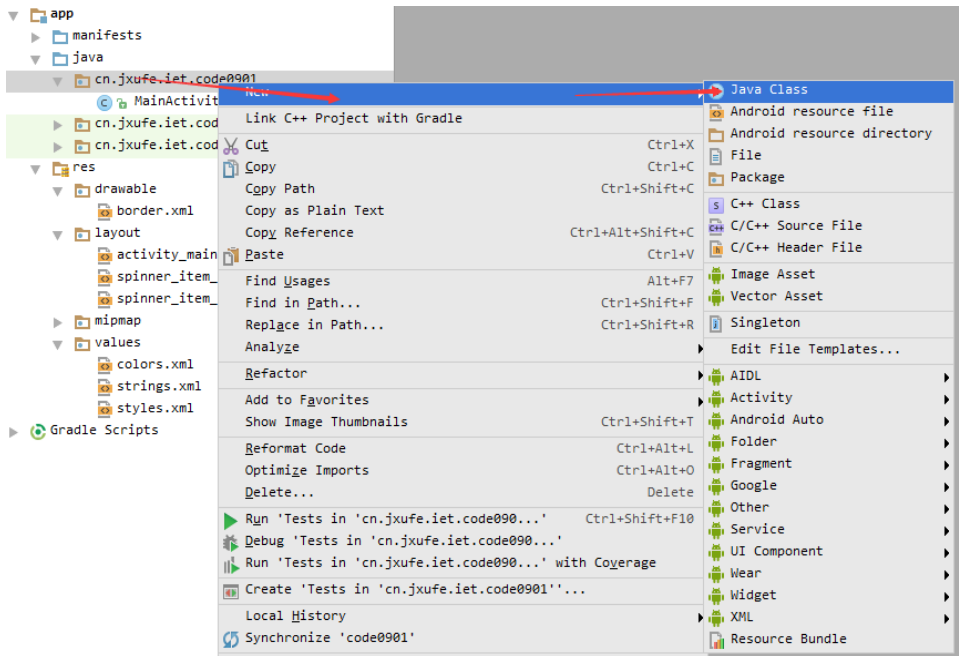


图9-4 新建“java class”

再新建 MyDatabaseHelper 类继承自 SQLiteOpenHelper 类，如图 9-5 所示。

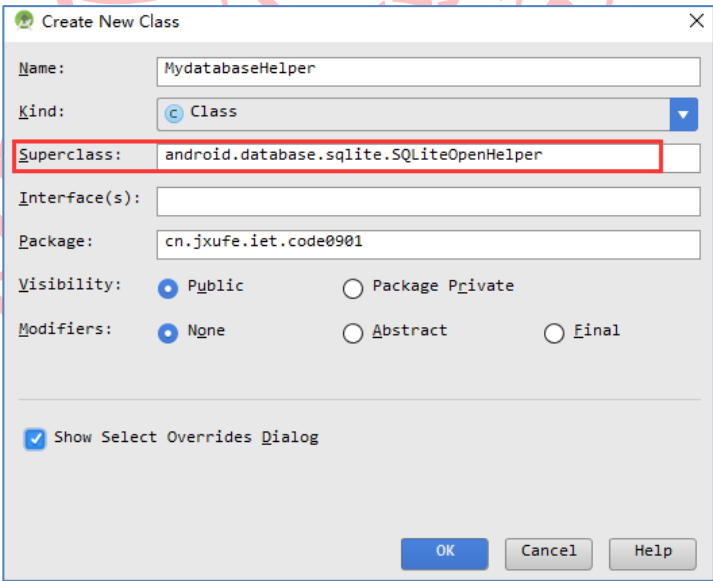


图9-5 新建“MyDatabaseHelper”类

打开 MyDatabaseHelper 类，将建表语句以字符串的形式写进类中，代码如下。

程序清单：Code0901\app\src\main\java\cn\jxufe\iet\code0901\MyDatabaseHelper.java

```

1 public class MyDatabaseHelper extends SQLiteOpenHelper {
2     //定义创建地区 area 表的 sql 语句
3     public static final String CREATE_AREA="create table area("
4         +"areaId integer primary key,"
5         +"areaName)";
6     //定义创建学校 school 表的 sql 语句
7     public static final String CREATE_SCHOOL="create table school("
8         +"schoolId integer primary key,"
9         +"schoolName text,"
10        +"schoolCode integer,"
11        +"areaId integer,"
12        +"is985 integer,"
13        +"is211 integer,"
14        +"isMinistry integer)";
15     private Context mContext;
16     public MyDatabaseHelper(Context context, String name,
17         SQLiteDatabase.CursorFactory factory, int version){
18         super(context,name,factory,version);
19         mContext = context;
20     };
21     public void onCreate(SQLiteDatabase db){
22         db.execSQL(CREATE_AREA);
23         db.execSQL(CREATE_SCHOOL);

```

可以看到我们把两张表的建表语句都定义成了字符串常量，然后在 onCreate() 方法中又调用了 execSQL 方法执行两条语句，这样可以保证在创建数据库的同时两张表也创建成功了。接下来我们打开 activity_main.xml 文件，修改布局文件如下所示：

程序清单：Code0901\app\src\main\res\layout\activity_main.xml

```

1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:id="@+id/activity_main"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical">
8     <LinearLayout
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:gravity="center_horizontal"
12        android:orientation="horizontal"
13        android:padding="5dp"
14        android:layout_gravity="center">
15        <LinearLayout
16            android:layout_width="wrap_content"
17            android:layout_height="wrap_content"
18            android:gravity="center_horizontal"

```


第 11 章 App 网络服务器管理程序设计

11.1 服务器开发背景知识

工欲善其事必先利其器，我们学习服务端的开发，就必须先要熟悉掌握开发的工具。现在比较流行的开发工具有 Eclipse、MyEclipse 和 WebBuilde 等。在这里，推荐大家使用 MyEclipse 软件进行服务端的开发，因为其整合了专门用于开发 Web 应用的各类框架和功能，例如 Spring 框架，Hibernate 框架等。本书使用的版本是 MyEclipse 2015。

11.1.1 JSP 简介

JSP (Java Server Pages) 是一种动态网页开发技术。这项技术是在 Servlet 基础之上开发的，继承了 Java Servlet 所有的优秀功能，使用 JSP 可以十分高效的创建和开发具有安全性高、可跨平台特点的 Web 应用程序。

11.1.2 Tomcat 服务器

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应 HTML (标准通用标记语言下的一个应用) 页面的访问请求。实际上 Tomcat 部分是 Apache 服务器的扩展，但它是独立运行的，所以当你运行 Tomcat 时，它实际上作为一个与 Apache 独立的进程单独运行的。也就是说，将你的项目放入到 Tomcat 服务器中，你就可以去访问你所开发的项目了。

我们所使用的 MyEclipse 2015 开发工具已经集成了 Tomcat 7.0 版本。但是，出于学习的目的，我们还是需要掌握如何去配置和搭建 Tomcat 服务器。

(1) 下载 Tomcat

首先，我们进入 Apache 官网 (<http://tomcat.apache.org/>) 去下载 Tomcat。如图 11-1 所示：

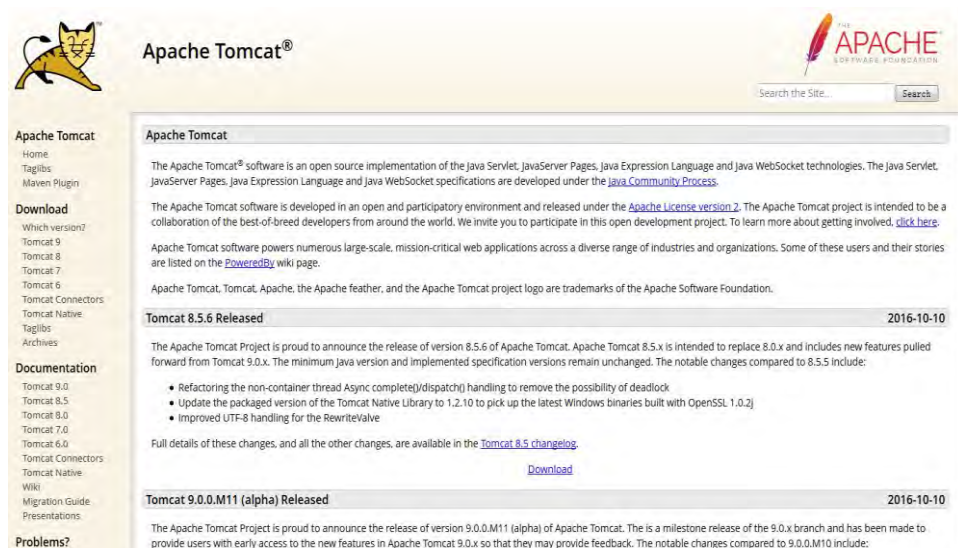


图11-1 下载Tomcat服务器步骤

找到官网左侧菜单栏 Download 栏目，选择你所需要的 Tomcat 版本，在这里，我们选 Tomcat 8 进行下载。如图 11-2 所示



图11-2 下载Tomcat服务器步骤

点击 Tomcat 8 进入，选择系统对应版本进行下载。如图 11-3 所示:

Servlet 通过创建一个框架来扩展服务器的能力,以提供在 Web 上进行请求和相应的服务。当客户端发送请求至服务器时,服务器可以将请求信息发送给 Servlet,并让 Servlet 建立起服务器返回给客户端的反应。当启动 Web 服务器或客户端第一次请求服务时,可以自动装入 Servlet,之后 Servlet 继续运行直到其他客户端发出请求,Servlet 是一个 Java 类,Java 语言能够实现的功能,Servlet 基本上都可以实现(图形界面除外)。总的来说,Servlet 是运行在服务器端的程序,用于处理及响应客户端的请求。

在 Servlet 的整个生命周期中,Servlet 的处理过程如图 11-20 所示:

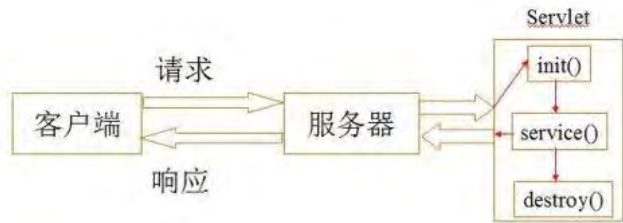


图 11-20 Servlet 的处理过程

图 11-20 所示各步骤的说明如下:

- (1) 用户通过客户端浏览器请求服务器,服务器加载 Servlet 并创建一个 Servlet 实例;
- (2) 容器调用 Servlet 的 `init()` 方法;
- (3) 容器调用 Servlet 的 `service()` 方法,并将 `HttpServletRequest` 和 `HttpServletResponse` 对象传递给该方法,在 `service()` 方法中处理用户请求;
- (4) 在 Servlet 中请求处理结束后,将结果返回给容器;
- (5) 容器将结果返回给客户端进行显示;
- (6) 当 Web 容器关闭时,调用 `destroy()` 方法销毁 Servlet 实例。

11.3.4 掌握 `HttpServletRequest` 类

`Request` 对象代表了客户端的请求信息,主要用于接受通过 HTTP 协议传送到服务器的数据。(包括头信息、系统信息、请求方式以及请求参数等)。`Request` 对象的作用域为一次请求。

当客户端向服务器端发送请求时,服务器为本次请求创建 `Request` 对象,并在调用 Servlet 的 `service` 方法时,将该对象传递给 `service` 方法。`Request` 对象中封装了客户端发送过来的所有的请求数据。

`Request` 对象获取请求参数的方法有以下几个,如表 11-1 所示

表11-1 request对象方法

方法名	方法介绍
String getParameter (String name)	获取指定名称的请求参数值, 适用于单值的请求参数
String[] getParameterValues (String name)	获取指定名称的请求参数值, 适用于多值的请求参数
Enumeration<String> getParameterNames ()	获取所有的请求参数名称
Map<String, String[]> getParameterMap ()	获取所有请求参数, 其中参数名作为 map 的 key, 参数值作为 map 的 value.

11.3.5 掌握 HttpServletResponse 类

Response 代表的是对客户端的响应, 主要是将服务器处理过的对象传回到客户端。Response 对象也具有作用域, 它只在 JSP 页面内有效。

表11-2 Response对象的常用方法函数

方法名	方法介绍
Response.write()	将指定的字符串写到当前的 HTTP 输出
Response.redirect ()	Redirect 方法可以使浏览器从当前网页转向到其他网页

11.3.6 实战演练

在这小节中, 我们将通过实际的小项目来加深对 Request 对象与 Responses 对象的理解。

1. 首先新建一个项目名为 code1101 的 Web 项目。
2. 在 src 文件夹下建立一个名为 drhelp.cn.servlet 的包, 右击 src 文件夹选择 New 再选择 Package。如图 11-21 所示

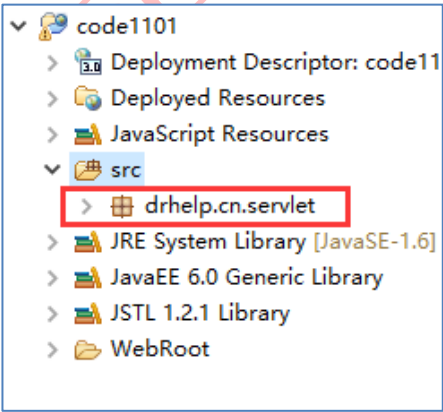


图11-21 新建drhelp.cn.servlet包

3. 新建 servlet 类。右击 drhelp.cn.servlet 包, 操作如图 11-22 所示

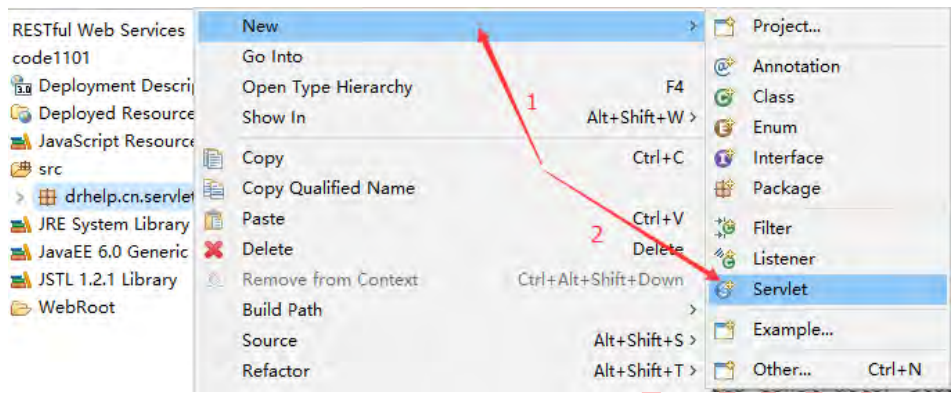


图11-22 新建servlet类

4. 点击 Servlet 后，弹出如下对话框，填写 Servlet 类名

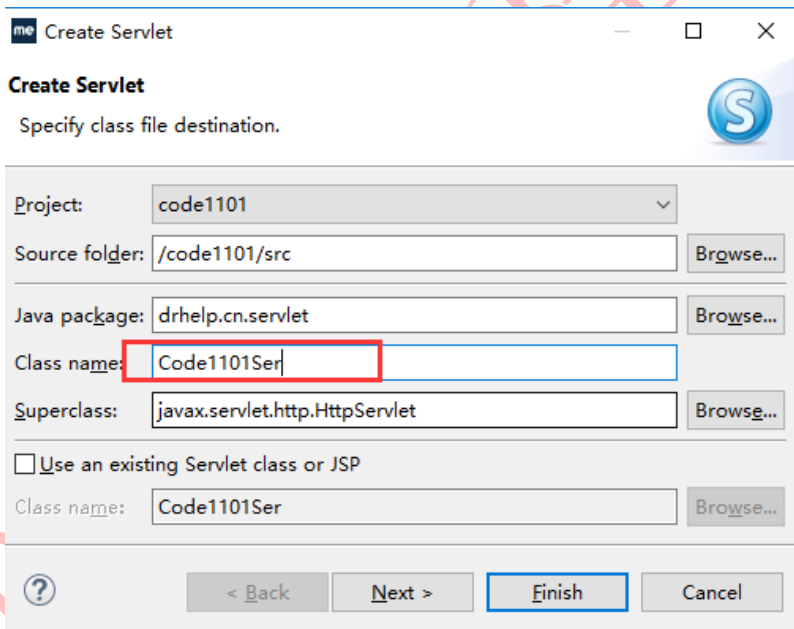


图11-23 填写Servlet类名

5. 点击 finish 后即可。进入创建好的 Servlet 类中，我们可以看到，这个 Servlet 有 `inti()` 方法、`destroy()` 方法、`goGet()` 方法和 `doPost()` 方法。Init 和 destroy 方法前面我们已经介绍过了。那么 `doGet` 方法和 `doPost` 方法函数分别是用来处理客户端提出的 Get 请求和 Post 请求。Get 请求和 Post 请求是指定了 `method=“Get”` 或者 `method=“Post”` 的表单时发出的请求。

我们在 `doPost()` 方法中定义一个 `String` 类型对象 `year` 用于存放客户端传递过来的值，再通过 `response` 对象将 `year` 传递给前台。

```

107         flag = true;
108     }
109     con.close();
110 } catch (SQLException e) {
111     e.printStackTrace();
112 }
113     return flag;
114 }
115 }

```

11.7.3 问题模块 Servlet 类

在问题模块的 **Servlet** 类中，我们都知道只有一个 **doPost** 或者 **doGet** 方法，可是我们问题模块具有三个功能，分别是提交问题、回复问题和查询问题。那么如何能够每一次都按需求自己选择执行对应的功能呢？其实要解决这个问题不难，我们只需要让 **android** 客户端也传递一个标识变量，然后在 **doPost** 或者 **doGet** 方法中使用 **if** 语句判断这个标识变量在运行相对应的程序代码即可。

首先在包 **drhelp.cn.servlet** 中新建一个问题模块的 **servlet** 类 **QuestionServlet**。

我们将代码写在 **doPost** 方法中，那么 **doGet** 方法里面就需要添加代码如下：

this.doPost(request, response)

doPost 中代码如下：

```

1  protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
2      response.setCharacterEncoding("UTF-8");
3      PrintWriter out = response.getWriter();
4      String flag=request.getParameter("flag");
5      QuestionImpl qi=new QuestionImpl();
6      if("insert".equals(flag)){//提交问题
7          Boolean result=qi.insertQuestion(request.getParameter("question"));
8          out.write(result+"");
9      }else if("update".equals(flag)){//回复问题
10         boolean result=qi.updateQuestion(request.getParameter("question"));
11         out.write(result+"");
12     }else{//查询问题
13         List<Question>
            questions=qi.getQuestions(request.getParameter("keyword"));
14         Gson gson=new Gson();
15         String result=gson.toJson(questions);
16         out.write(result);
17     }
18 }

```

首先定义一个 **String** 类型 **flag** 来存放 **android** 客户端传递过来的标识变量（值为 **query** 或者 **null**，默认即为查询；**update** 为回复问题；**insert** 为提交问题），我们使用 **if** 语句来对 **flag** 进行判断，并在 **if** 里面调用不同的方法进行不同的操作。

到这里，我们问题模块的 **Servlet** 类也编写好了。接下来，我们将项目部署到

第 12 章 App 客户端与服务器交互设计

前面章节中，已经实现了用本地临时数据的“豹考通”客户端和搭建好的服务端，客户端还没有和服务端进行交互。本章主要实现客户端替换临时数据与服务端进行交互，其中包括客户端和服务端交互理论、“省控线查询”模块与服务端交互实现过程、“历年录取线查询”模块与服务端交互实现过程和“报考咨询”模块与服务端交互实现过程。

12.1 客户端和服务端数据交互基础

12.1.1 HttpClient

Android 客户端与服务器端交互方式通常有两种：一种是直接采用 Android 内置的 HttpClient 发送 HTTP 请求和获取 HTTP 响应；另一种是利用 ksoap2-android 项目，调用服务器端提供的 WebService。“豹考通”项目中采用第一种方式，通过 HttpClient 与服务器端进行交互。

HttpClient 是 Apache 开源组织提供的一个项目，HttpClient 开发过程中所涉及类主要包含以下几个。

- **HttpClient:** HTTP 客户端接口，该接口封装了执行 Http 请求所需要的各种对象，这些对象可以处理 cookie、授权、连接管理以及其他一些特性。HttpClient 的线程安全依赖于具体的客户端的实现和配置。该接口中包含多种 execute() 方法，用于执行具体的请求，通常通过 DefaultHttpClient 子类来创建该接口的对象。
- **HttpGet:** 该类采用 GET 方式发送 HTTP 请求，通常将请求的 URL 作为参数传给该类的构造方法；
- **HttpPost:** 该类采用 POST 方式发送 HTTP 请求，通常将请求的 URL 作为参数传给该类的构造方法；
- **HttpResponse:** 该接口封装了 HTTP 的相应信息，通过调用相应方法可以获取 HTTP 响应信息，例如获取响应状态、响应内容等。
- **HttpEntity:** HTTP 信息封装类，通过该类可以获取 HTTP 请求或响应的内容、长度、类型、编码方式等信息，HttpEntity 对象可以通过 Http 消息发送和接收，既可以存在于请求消息中，也可以存在于响应消息中。

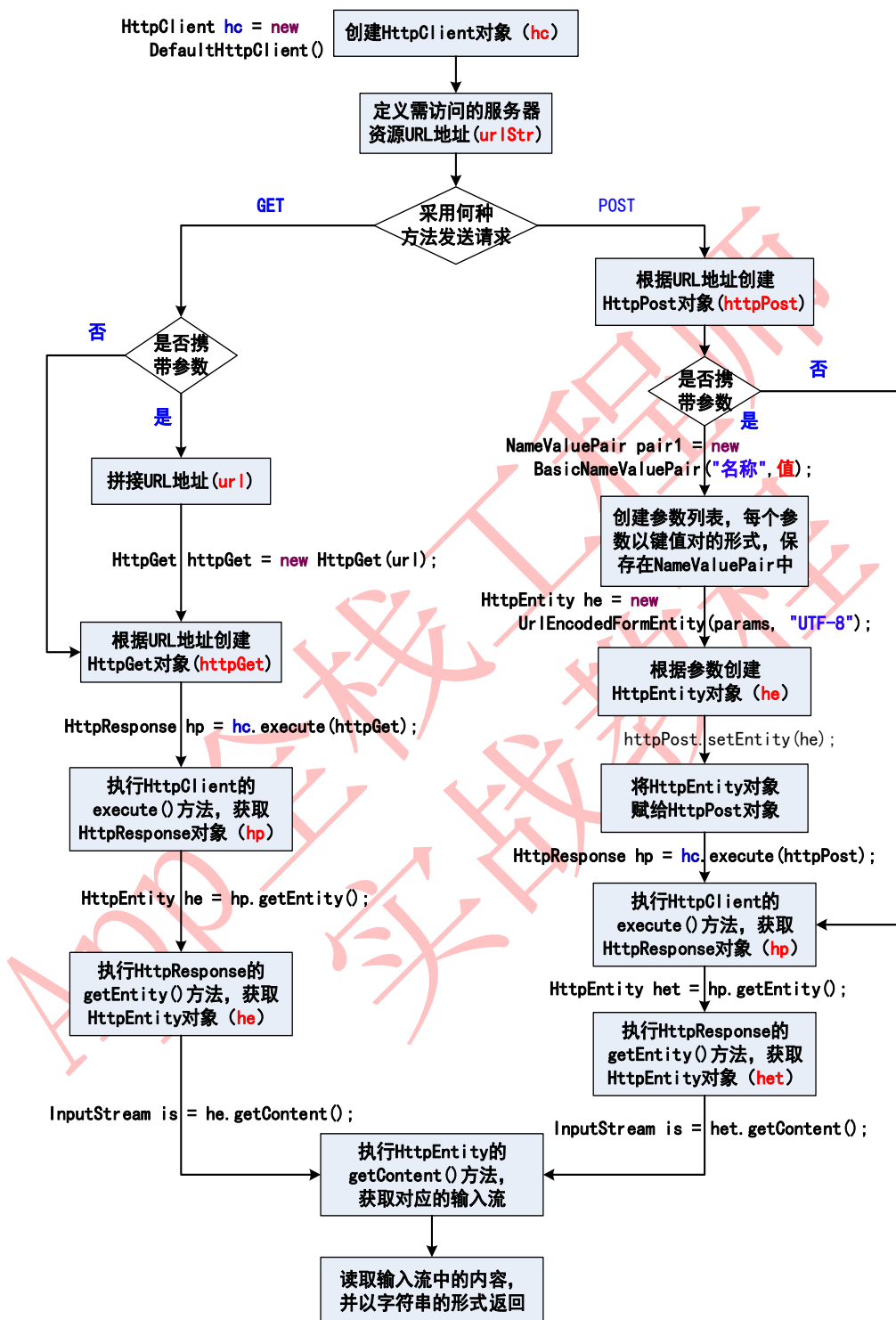


图12-1 HttpClient发送请求的过程

这些类又是如何协同工作共同来完成 Http 请求和获取响应信息的呢？通常会采用 GET 和 POST 两种方式发送请求，而 GET 和 POST 发送请求各有优缺点，需根据具体的情境进行选择，两者的操作模式有所差别，主要是参数传递方式有所不同。因此，在通过 HttpClient 与服务器端交互时，需要针对这两种情况单独处理。使用 HttpClient 发送请求获取响应信息的流程如图 12-1 所示。

通过图中所述执行过程，我们可以总结出无论是使用 HttpGet，还是使用 HttpPost，都必须通过以下 3 步来访问 HTTP 资源。

(1) 创建 HttpGet 或 HttpPost 对象，将要请求的 URL 通过构造方法传入 HttpGet 或 HttpPost 对象。

(2) 使用 DefaultHttpClient 类的 execute 方法发送 HTTP GET 或 HTTP POST 请求，并返回 HttpResponse 对象。

(3) 通过 HttpResponse 接口的 getEntity 方法返回响应信息，并进行相应的处理。

注意：如果使用 HttpPost 方法提交 HTTP POST 请求，还需要使用 HttpPost 类的 setEntity 方法设置请求参数。访问网络时，需要在清单文件中添加访问网络的权限，此外，在 Android 高版本中（Android 3.0 以上），将访问网络看做是一种较为耗时的操作，不能直接在主线程中访问网络，需要单独启动线程或通过异步任务来访问网络。

12.1.2 网络服务器端返回结果解析—JSON 解析

服务器端响应客户端请求返回的结果的表示方式有多种，既可以是具体的 HTML 页面、也可以是存储结构化数据的 XML 文件，也可以是 JSON 格式的字符串等等。“豹考通”项目中，客户端与服务器端数据交互采用 JSON 格式，通过谷歌提供的第三方 Jar 包—Gson 来实现 JSON 格式字符串的生成和解析。

JSON (JavaScript Object Notation) JavaScript 对象符号，是一种轻量级的数据交换格式，JSON 采用完全独立于语言的文本格式，既易于人阅读和编写，同时也易于机器解析和生成。JSON 中主要有两种数据结构。

- 由 key-value 对组成的数据结构，这种结构在不同语言中有不同的实现，例如在 C 语言中是一个 struct，在 Java 中是一种 Map 结构，在 JavaScript 中则是一个对象等。
- 有序集合，这种数据结构在不同语言中，可能是列表、集合、数组等实现。

“豹考通”项目中使用 Google 提供的 Gson 来实现 Java 对象和 JSON 数据之间转化，Java 对象与 JSON 数据之间的转化主要包含两种：一种是将 Java 对象转换成 JSON 格式，只需要调用 Gson 对象的 toJson() 方法，并将 Java 对象作为参数传递即可；另一种是将 JSON 格式的字符串转换成一个 Java 对象，也非常简单，只需要调用 Gson 对象的 fromJson() 方法，传递两个参数，一个是需要转换的 JSON 格式的字符串，一个是需要转换的对象的类型。详细代码会在后面的具体模块实现部分列出。

12.2 本章案例实现流程

本章是在前面开发本地客户端和网络服务器开发完成的基础上进行完成。首先，在前面开发的客户端基础上把第三方 jar 包导入；其次，在 util 文件夹下，创建一些与服务端交互用到的工具类；然后，分别实现与服务端交互的“省控线查询”、“历年录取线查询”和“报考咨询”三个模块，具体流程图请参考图 12-2。



图12-2 实现“客户端与服务端交互”的开发流程

12.3 第三方 Jar 包导入

“豹考通”客户端项目中主要用到了两个第三方 Jar 包：gson-2.3.1.jar 和 org.apache.http.legacy.jar 包。导入第三方 jar 包有两种方式，具体如下。

方式一：

1. 下载需要的 jar 包。
2. 将 jar 包复制到 Project 下的 app ->libs 目录下（没有 libs 目录就新建一个）如下图 12-3 所示位置。

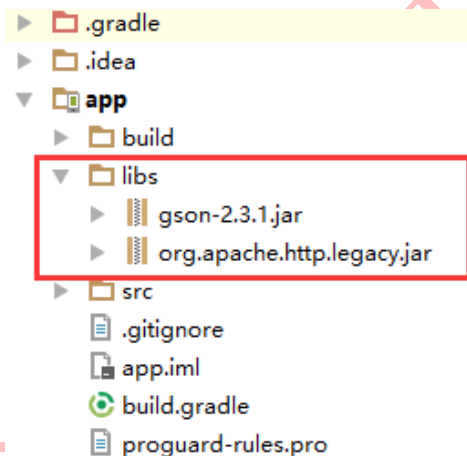
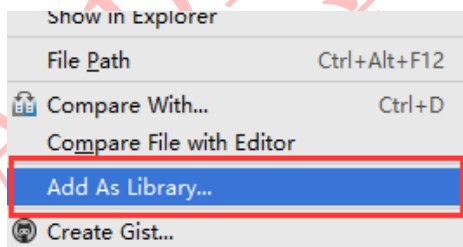
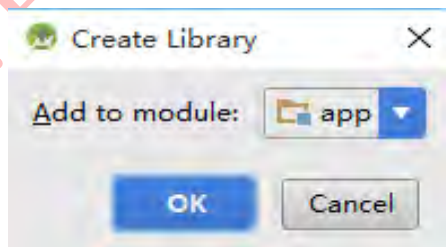


图12-3 复制Jar包到libs文件夹下

3. 右键该 jar 包，选择 add as library，添加依赖包，弹出如下窗口，点击 ok 即可，如图 12-4 所示。



(a) add as library



(b) Add to module

图12-4 添加依赖包

4. 打开 build.gradle 文件，在 dependencies 方法中出现 Jar 包名称，说明导入成功，如图 12-5 所示。

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.4.0'  
    compile files('libs/gson-2.3.1.jar')  
    compile files('libs/org.apache.http.legacy.jar')  
}
```

图12-5 Jar包导入成功

方式二：

1. 下载需要的 jar 包
2. 将 jar 包复制到 Project 下的 app ->libs 目录下（没有 libs 目录就新建一个）
如图 12-6 所示位置。

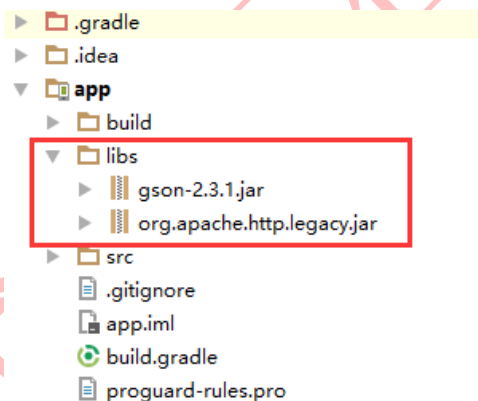


图12-6 复制Jar包到libs文件夹下

3. 点击工具栏中的 Project Structure 按钮，如下图 12-7 所示。

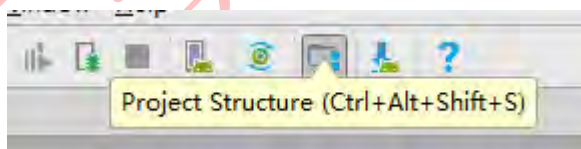


图12-7 点击 Project Structure按钮

4. 选择 Dependencies, 点击右边的加号, 选择第二项 Jar Dependency, 如下图 12-8 所示。

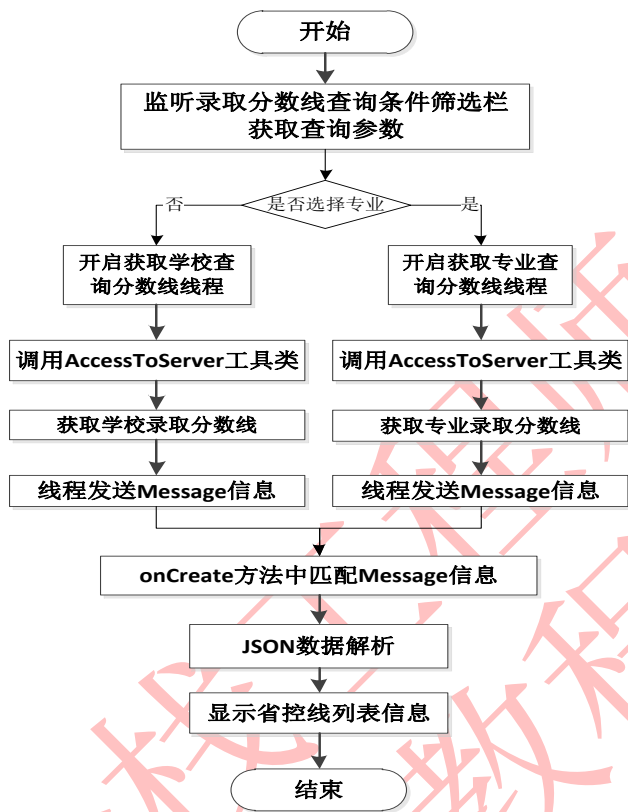


图12-16 “历年录取线查询”模块实现具体流程

12.6.2 获取服务端学校录取线和专业录取线

在 EnrollScoreFragment 类定义常量，Handler 和 Gson 常量，代码如下。

程序清单：Code1202\app\src\main\java\cn\jxufe\iet\code1202\fragment\EnrollScoreFragment.java

```
1 private Handler mHandler;
2 private boolean isFinishedInit = false;//是否完成初始化
3 private int schoolServiceId;//学校在服务端数据库中 Id
```

事件监听条件筛选栏发生一些改变，因为由原来本地临时数据转换成从服务端获取数据，请求服务端数据时需要传递查询学校在服务端的 Id 信息。所以在事件监听器类 MyItemSelectedListener 做一些简单的调整，在“case R.id.schoolSpinner:”下添加“schoolServiceId = schoolList.get(position).getSchoolId();// 获取学校在服务端数据库中 Id”代码，如图 12-17 所示。

程序清单：Code1203\app\src\main\java\cn\jxufe\iet\code1203\fragment\ConsultFragment.java

```

1  @Override
2  public void onResume() {
3      super.onResume();
4      getHistoryQuestion(keywordView.getText().toString().trim());
5  }

```

从服务端获取反馈历史问题 JSON 数据后，需要把数据 JSON 数据解析并以列表的形式显示。重写 onCreate 方法中，匹配发送信息后，解析 JSON 数据后调用 showQuestions 方法显示列表数据，代码如下。

程序清单：Code1203\app\src\main\java\cn\jxufe\iet\code1203\fragment\ConsultFragment.java

```

1  public void onCreate(Bundle savedInstanceState) {
2      super.onCreate(savedInstanceState);
3      mHandler = new Handler() {
4          @Override
5          public void handleMessage(Message msg) {
6              String result = (String) msg.obj;
7              result = (String) msg.obj;
8              switch (msg.what) {
9                  case Constants.GET_QUESTIONS:
10                     if ("exception".equals(result) || "error".equals(result)
11                         || "".equals(result)) {
12                         Toast.makeText(getActivity(), "访问服务器异常!",
13                             Toast.LENGTH_SHORT);
14                         return;
15                     }
16                     questions = gson.fromJson(result,
17                         new TypeToken<List<Question>>() {
18                             }.getType());
19                     if (questions != null) {
20                         showQuestions();
21                     }
22                     break;
23                 default:
24                     break;
25             }
26         }
27     }; // 初始化 Handler 对象
28 }

```

在上面代码中，我们发现 showQuestions 方法还没有实现，showQuestions 方法是显示问题列表，具体代码如下。

程序清单：Code1203\app\src\main\java\cn\jxufe\iet\code1203\fragment\ConsultFragment.java

```

1  public void showQuestions() { // 显示问题列表
2      ConsultAdapter consultAdapter=new
3      ConsultAdapter(getActivity(), questions);
4      questionListView.setAdapter(consultAdapter);
5  }

```


12.7.3 “提问”模块实现

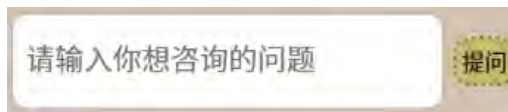


图12-21 “提问”界面

点击“提问”按钮，能够把咨询编辑框内输入的内容，提交到服务端，并能更新问题列表数据，如图 12-21 所示。首先需要对提问按钮匿名内部类事件监听，并把输入的问题封装成 JSON 数据格式，便于后面提交到服务端，具体代码如下。

程序清单：Code1203\app\src\main\java\cn\jxufe\iet\code1203\fragment\ConsultFragment.java

```

1  submit.setOnClickListener(new View.OnClickListener() { // 提交按钮的事件处理
2      public void onClick(View v) {
3          String questionContent = contentView.getText().toString()
4              .trim(); // 获取输入的问题
5          if ("".equals(questionContent)) {
6              AlertDialog.Builder builder = new AlertDialog.Builder(
7                  getActivity());
8              builder.setMessage("请输入咨询的问题，内容不能为空!"); // 对话框显示的内容
9              builder.create().show(); // 创建并显示对话框
10         } else {
11             Question question = new Question();
12             question.setQuestionContent(contentView.getText().toString());
13             question.setQuestionTime(Util.getDate());
14             questionString = gson.toJson(question);
15             submitQuestion();
16         }
17     }
18 }
19 }); // 提问按钮的单击事件处理

```

提交问题获取并封装成 JSON 数据后需要把数据提交到服务端，在上面代码我们也发现 submitQuestion 方法还未实现，这时需要实现 submitQuestion 方法，并通过调用 AccessToServer 工具类把数据提交到服务端，具体代码如下。

程序清单：Code1203\app\src\main\java\cn\jxufe\iet\code1203\fragment\ConsultFragment.java

```

1  public void submitQuestion() {
2      if (Util.isNetworkConnected(getActivity())) {
3          new Thread() {
4              public void run() { // 线程执行体
5                  AccessToServer accessToServer = new AccessToServer();
6                  String result = accessToServer.doPost(Global.URL
7                      + "QuestionServlet", new String[] { "question",

```

第 13 章 App 集成测试

13.1 单元测试

单元测试，是指对软件中的最小可测单元进行检查和验证。对于单元测试中单元的含义，一般来说，要根据实际情况去判定其具体含义，如 C 语言中单元指一个函数，Java 中单元指一个类，图形化的软件中可以指一个窗口或一个菜单等，总的来说，单元就是人为规定的最小的被测功能模块。单元测试是在软件开发过程中要进行的最低级别的测试活动，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。单元测试主要以白盒测试为主，夹杂着少量的黑盒测试。

13.1.1 黑盒测试

黑盒测试也称功能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。下面以“历年高考省控线查询”模块为例。



图13-1 “历年高考省控线查询”界面图

输入年份查询省控线，如图 13-2、图 13-3、图 13-4 和图 13-5 所示。



图13-2 查询省控线结果



图13-3 查询省控线结果



图13-4 查询省控线结果



图13-5 查询省控线结果

由上图测试可知，“历年高考省控线查询”模块功能可正常实现。

13.1.2 白盒测试

白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，你清楚盒子内部的东西以及里面是如何运作的。“白盒”法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。此处用语句覆盖（每条语句至少执行一次）的方式对“历年高考省控线查询”模块进行测试。

（1）运用数组实现“历年高考省控线查询”的核心类代码如下所示。

```

1.  @SuppressWarnings("HandlerLeak")
2.  public class demo extends Activity {
3.      private String year;
4.      private TextView tv;
5.      private Button bl;
6.      private EditText et;
7.      @Override
8.      protected void onCreate(Bundle savedInstanceState) {
9.          // TODO Auto-generated method stub
10.         super.onCreate(savedInstanceState);
11.         setContentView(R.layout.demo);
12.         bl=(Button) findViewById(R.id.button);
13.         tv= (TextView) findViewById(R.id.result1);
14.         et=(EditText) findViewById(R.id.input1);
15.         bl.setOnClickListener(new OnClickListener() {
16.             @Override
17.             public void onClick(View v) {
18.                 // TODO Auto-generated method stub
19.                 year=et.getText().toString();
20.                 System.out.println(year);
21.                 getControlLine();
22.             }
23.         });
24.     }
25.     public void getControlLine() {
26.         boolean flag=true;
27.         String[][] data={{"2013","617"}, {"2014","630"}};
28.         for(int i=0;i<data.length;i++){
29.             if(year.equals(data[i][0])){
30.                 flag=false;
31.                 String result=data[i][1];
32.                 tv.setText(result);
33.             }
34.         }
35.         //无数据显示暂无数据
36.         if(flag){
37.             tv.setText("暂无数据");
38.         }
39.     }
40. }

```

首先，我们需要设计用例，用例如表 13-1 所示。

表13-1 历年高考省控线查询测试用例

用例编号	输入数据	预期输出
1	year = 2013	617
2	year = 2014	630
3	year = 2015	暂无数据

然后，打开 Android Studio，初始化界面如图 13-6 所示。



图13-6 Android Studio开始界面

向 Android Studio 中导入 bktDemo（本地数据版历年高考省控线查询模块）。向“历年高考省控线查询”的核心类代码中加入三条测试语句（7, 33, 39 三行，加粗部分），代码如下所示。

```
1. @SuppressWarnings("HandlerLeak")
2. public class demo extends Activity {
3.     private String year;
4.     private TextView tv;
5.     private Button b1;
6.     private EditText et;
7.     private static final String TAG = "demo";//定义以当前类名为值的 TAG 常量
8.     @Override
9.     protected void onCreate(Bundle savedInstanceState) {
10.         // TODO Auto-generated method stub
11.         super.onCreate(savedInstanceState);
12.         setContentView(R.layout.demo);
13.         b1=(Button) findViewById(R.id.button);
```



图13-7 历年高考省控线查询主界面

接下来，我们需要创建日志过滤器来过滤掉一些不必要的日志信息，点击 6:Android Monitor，将出现如图 13-8 所示界面。

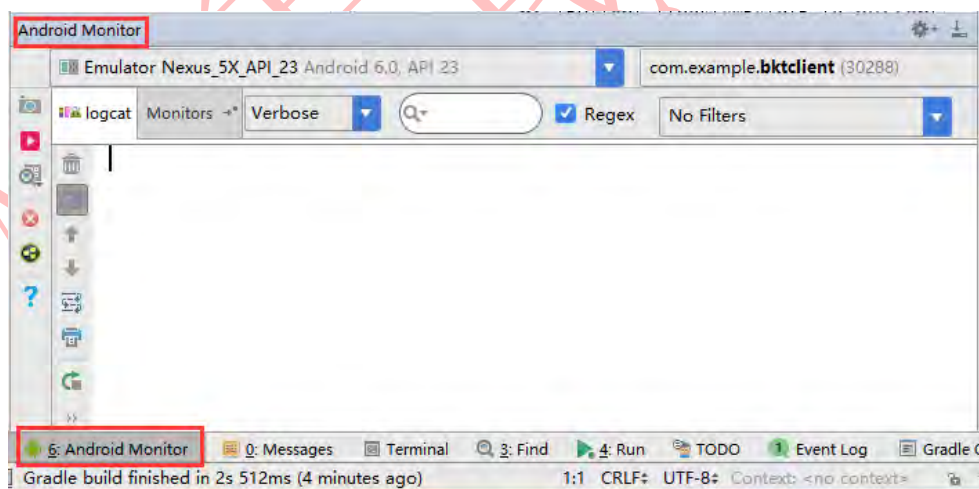


图13-8 Android Monitor界面

我们在 Regex 下拉菜单中选择 Edit Filter Configuration，如图 13-9 所示。

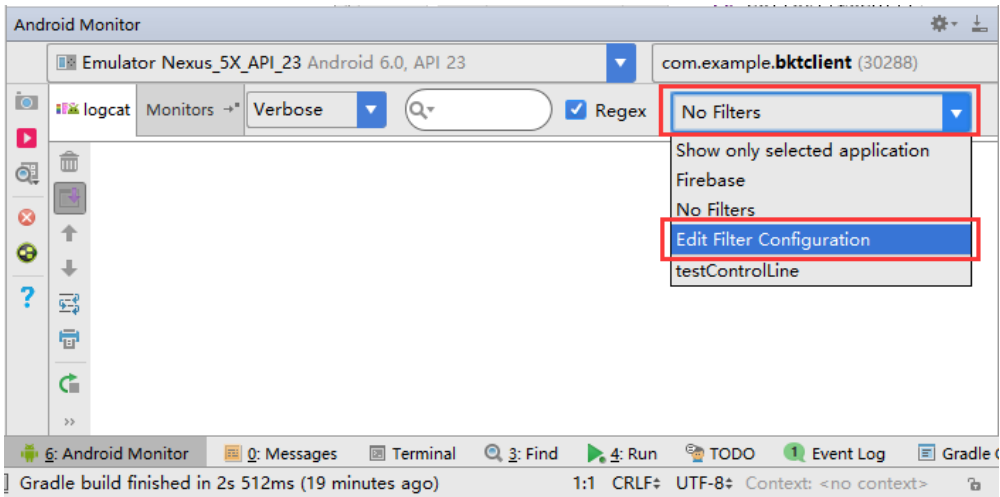


图13-9 Android Monitor界面

选择 Edit Filter Configuration 之后,将出现如下对话框,我们在 Filter Name 中输入 testControlLine, Log Tag 中输入 demo, Log Level 下拉菜单中选择 Warn, 然后点击 OK。如图 13-10 所示。

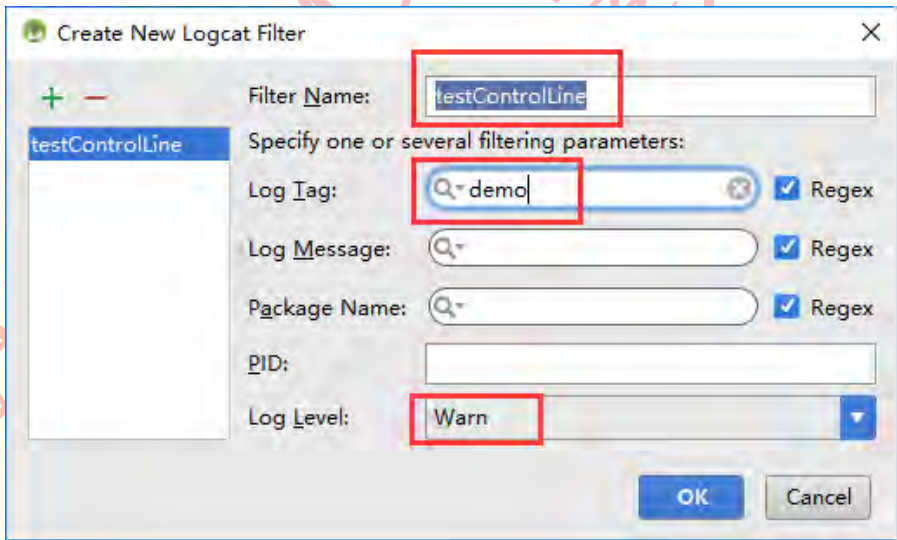


图13-10 Create New Logcat Filter界面

最后,我们在模拟器界面上依次输入 2013, 2014 和 2015 三个年份来查询省控线。如图 13-11, 13-12, 13-13 所示。

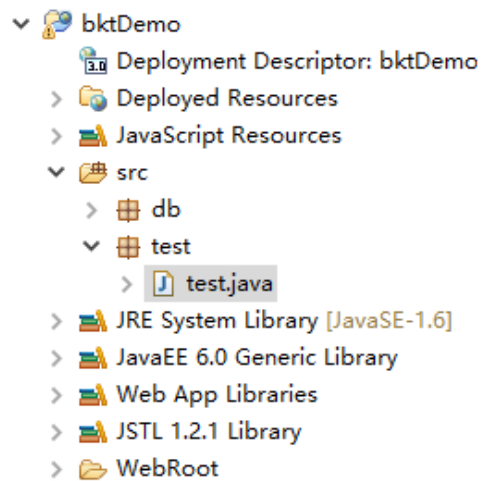


图13-15 程序目录结构

test.java代码如下:

```
1. package test;
2. import db.DBconn;
3. public class test {
4.     public static void main(String[] args) {
5.         DBconn dbc=new DBconn();
6.         dbc.getConnection();
7.     }
8. }
```

新建 DBconn 对象,再调用 DBconn 对象的 getConnection 方法。运行 test. java 文件,结果如下图 13-16 所示。

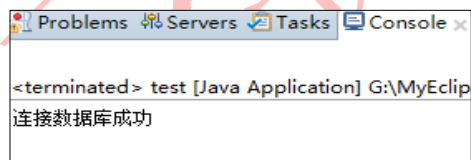


图13-16 数据库连接类测试结果

13.2.2 Jave Web 服务器访问 MySql 数据库集成测试

在第3章设计网络服务器小节中,我们在MyEclipse中编写好了数据库连接类,存放数据用的实体类,实施数据库操作的工具类,其目录结构如图 13-17 所示。

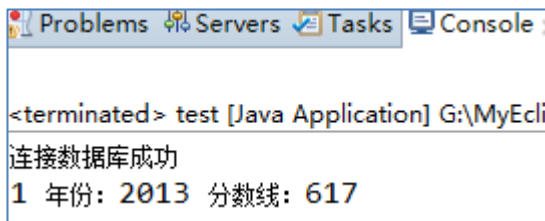


图13-18 数据库操作类实现结果

上面的结果还不完全是 Java Web 服务器访问 MySQL 数据库的结果。还需要我们实现 Servlet 类。Servlet 类是 Web 服务器或客户端第一次请求服务器时运行的 java 类。在我们服务器中，我们使用 Gson 来解析 JSON 数据，Gson 是 Google 提供的用来在 Java 对象和 JSON 数据之间进行映射的 Java 类库，它可以将一个 JSON 字符串转换成一个 java 对象，也能够将一个 java 对象转换成 JSON 字符串。上面步骤全部完成之后，我们就编写好了服务器端。接下来我们需要将项目部署到 Tomcat 服务器中，部署完之后，启动 Tomcat 服务器，打开浏览器，在地址栏输入 `http://localhost:8080/bktDemo/LineSer?controlYear=2013`，这就是 android 客户端查询省控线的接口。简单理解的话就是向服务器传递年份 2013。

输入 URL 之后如果出现如图 13-19 所示结果，那么服务器端就算成功完成了。也就说明我们的 Java Web 服务器能够成功访问到 MySQL 数据库中的数据。

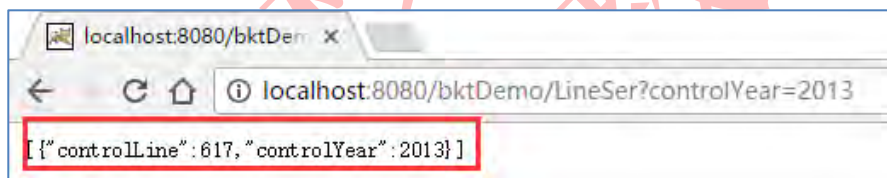


图13-19 测试结果

13.2.3 Android 客户端访问 Java Web 服务器集成测试

由上述 13.2.2 小节可知，我们的 Java Web 服务器已经能够成功地访问到 MySQL 数据库中的数据，那么我们能否将查询到的数据传输给安卓客户端呢？

我们需要在 Android 端，建立实体类来存放从服务器端传递过来的 JSON 数据。建好实体类之后，还需要建立一个工具类用来访问服务器和获取服务器返回的结果。上述步骤全部完成之后，我们将 Android 客户端运行起来，其主界面如图 13-20 所示。



图13-20 运行客户端主界面

我们在查询框中依次输入 2013、2014 两个年份来查询省控线，目的是测试 Android 客户端能否成功访问 Java Web 服务器，其执行结果如图 13-21、13-22 所示。



图13-21 测试结果



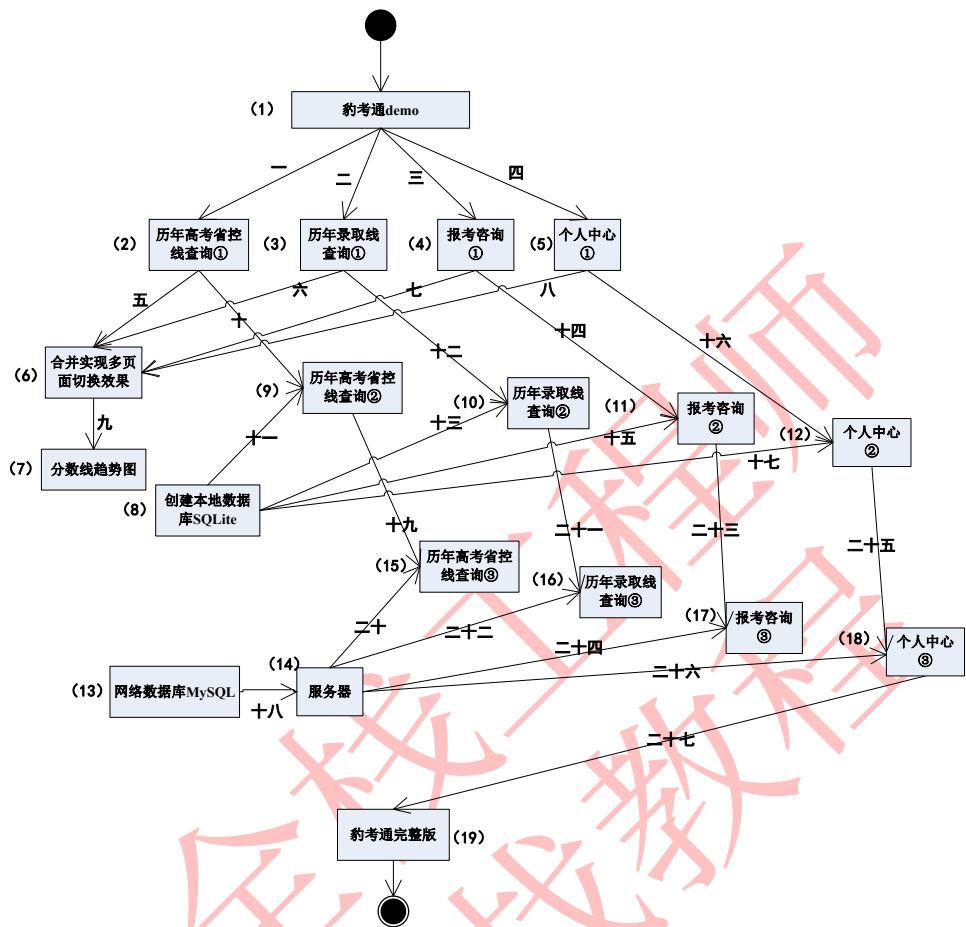
图13-22 测试结果

由上述测试结果可知,我们已经在 Android 客户端成功访问到 Java Web 服务器。

13.3 “豹考通”案例集成测试流程

13.3.1 集成测试流程

“豹考通”案例集成测试流程如图 13-23 所示。



其中 (1)、(2)、(3)、... 等序号代表依次集成顺序，一、二、三、... 等序号代表操作行为。

豹考通 demo 是集成测试的第一个基础，它实现了“历年高考省控线查询”的功能（界面如图 13-24 所示）。

第 14 章 App 全栈工程师创业思维训练

作为本书内容的提高版，本章介绍了我们在软件工程（移动商务方向）本科生中开展的一个培养“互联网+”创业人才的教学案例，供对“互联网+”创新创业有兴趣的师生选读，意在抛砖引玉，以期进一步提升 App 全栈工程师的价值。

14.1 App 全栈工程师创业思维训练的价值

14.1.1 有利于 App 全栈工程师自身的长远发展

1. 通过分析计算机人才培养以及许多程序员成长过程，我们很容易发现绝大部分程序员更强调算法的研究以及实施确定性的商业方案，对互联网创业规律不熟悉而错过很多好机会。

主要表现有：

- （1）以求得稳定的编程工资或外包费用为主，不敢创业，怕冒险。
- （2）模仿式的创新为主，项目定位不清，商业意识不强，盲目参与创业。
- （3）技术研发习惯解决确定性的需求，不了解互联网创业常常通过早期产品来获取用户需求、产品满足新需求不断迭代的办法，并因此而与创业团队有较大的抵触，影响创业项目的成长。
- （4）在面临颠覆式创新机会时，即使自己掌握了这一机会，也怕大公司介入而不敢创业，错失良机。
- （5）因不懂互联网创业股权和期权等方面的知识，要么过于乐观，要么过于保守，导致创业项目后期因股权、期权等方面的矛盾难以解决，严重影响项目的发展。

2. 当前我国倡导大学生“互联网+”创新创业，而大学开设的创业课程常常面向所有专业的，并不针对程序员开设。对于程序员来说，这类课程指导性不强，起不到应有之作用。

主要表现有：

- （1）因大学生专业知识所限，互联网+创新创业从项目选择开始就走很多弯路，浪费学生宝贵的时间，还可能因时间错过而直接放弃。
- （2）大学生不了解有什么好的创新技术可用于互联网+创业，不熟悉其应用领域，而导致大学生思路难以打开，感觉创业无从下手。
- （3）因为指导老师多数也没有创业经验，存在前述程序员创业的通病，也无法有效指导大学生创业。
- （4）现在的创业课程缺乏好的技术体验内容，很多已经有课程常常技术太深而难以让学生快速了解，学生常转而选择简单的已有商业形态的项目。

3. 互联网+创业公司常有开发 App 要求, 如互联网+具体应用的 App, 但大学生互联网+创新创业教育更是希望短平快的可模仿课程, 目前这类课程资源很少。特别应用物联网技术、人工智能技术的创业项目, 在许多大学里面更没有相应的实践课程。这就导致大学生互联网+创业实践中, 创意难以落地。

14.1.2 有利于高校开展“互联网+”双创人才培养工作

通过开展有别于不分专业的大众化的创业课程中的学习与训练, 面向 App 程序员群体, 可以更好地推动不同院校的大学生“互联网+”创新创业工作的有效开展。

这类学习与训练有助于程序员更清楚了解互联网+创业项目的选择, 能区分颠覆式创新与改进性创新对于创业的影响, 掌握精益创业的基本常识, 明确互联网+创业公司的股权、期权常识, 理解互联网+创业公司的发展规律, 使程序员能更好地抓住机会、勇于创业、善于创业, 作一个好的创业项目的技术合伙人。

大学生可以更早地体验以物联网、人工智能、大数据为基础的互联网+创业项目的应用以及发展前景, 启发大学生去思考、去设计基于这些技术的商业应用。

这种训练课程, 有利于解决当前普通大学的学生互联网+创新创业教育落地难的问题。随着大学生全国互联网+创新创业活动的不断推进, 许多大学生也都想将自己的好点子, 用 App 实现, 因为没有好的课程与教学案例可供学习而进展困难。特别是在许多非重点高校, 情况尤其严重。

总之, 这部分内容和课程的建设有利于在大学生中普及互联网+创业知识, 提高 IT 类大学生开展互联网+创业的综合能力, 也有利于帮助培养适合于“互联网+”初创公司的程序员, 还有利于提高计算机类大学生的就业能力。

14.2 App 全栈工程师创业思维训练基础

绝大多数程序员参与创业, 都不是自己当老板。多数大学生在校或刚走向社会时, 创业的条件是不成熟的。但是, 掌握一些创业常识, 还是很必要。

14.2.1 掌握定位理论, 选好创业项目

选择常常重于努力, 会不会选很重要。是否加入一家创业公司, 或者启动一个创业项目, 是首要问题。它决定了你将来几年的精力投入和收入保证, 影响你的长远发展。即使一个成熟项目或成熟公司, 也可问问其前景如何? 我有加入必要吗? 此时, 定位思想很有用。

定位理论源自营销定位, 我们认为它对于创业也有指导意义。通过网络搜索或看营销图书, 读者很方便地读到相关知识。本书省略理论介绍, 提出几个最基本的问题(见表 14-1)供程序员思考, 以培养其创业思维。

实际上, 定位理论作用广泛, 对一个人的定位、一个地区的定位, 乃至一个国家的定位都是有参考意义的。

表14-1 创业项目定位的若干思考点

序号	思考问题	定位关键点	应避免的误区
1	我提供什么产品或服务?	有什么功能或作用,即功能定位。	同别的产品功能差不多,或者什么功能都想要,追求处处比人优、比人强。
2	我为谁服务?	满足某一类人,或人的某一类需求,即消费者定位。	试图讨好每个人,变成大家都不满意;试图服务人的各方面需求,导致哪方面都不突出,没人买帐。
3	我赚人哪方面的钱?	产品为社会提供了什么价值,人们愿意花钱购买具有这一价值的产品,即价值定位。	谁的钱都想赚;或者人们的各方面钱你都想赚。价值不大,或有其他渠道能满足且价格更低,可选更便捷的方法。
4	为什么是我?	差异化定位:产品、用户、渠道等之一或多个方面我有独特之处;资源匹配:我能干而别人不能干或不愿意干,我有什么不可替代的优势。	做太多,干自己的能力或资源无法满足的事;别人能行,我也行;市场调查不充分,过于乐观,困难估计不足。

14.2.2 了解创新者的窘境,思考创业项目的创新路径

无创新不可能创好业,“互联网+”创业更是以创新推动创业。但是思考创新路径,对于很多程序员来说是完全没有思考的,绝大部分人都仅从更多、更优、更强等因素出发,为创新而创新,而没把创新放到创业的背景下来开展。表14-2列出了创业时对于创新路径的选择时需要思考的一些问题。

表14-2 创业项目创新路径的若干思考点

序号	思考问题	创新路径关键点	应避免的误区
1	我要创造什么新产品?	用有新功能的产品替代现有产品还是有明显价格优势的产品替代同样功能的老产品,还是创造老产品不能满足边缘市场需求的新产品。	当全新产品目标是老产品完全忽视的边缘市场用户需求时,过于强调其完美程度,不敢早日推向市场,或者一开始就希望全新产品会被主流市场认可。
2	我的产品有何技术创新?	我的创新是走技术改进道路,还是实现颠覆性技术。产品是价廉物美,还是市场空白?	以改进性创新的思想看待颠覆性创新技术,持悲观态度,瞧不起新机会,错失时机;老想着颠覆式的破坏创新,追风口,不屑于持续性的改进创新。
3	我的小众市场在哪里?	用新技术解决主流市场未满足的用户痛点,这些痛点老技术无法解决或解决不经济。	只调查现有主流市场的用户需求,得到不可行的结果,或者只考虑将新技术用于解决老产品的问题。
4	我的产品怎么占领主流市场?	引导需求,从边缘走向主流。通过持续改进,将价很廉、物不够美的,变成价廉物美;将填补空白的价高的,提质、降价、覆盖主流产品功能。	信心不足,缺乏改进创新行动。认为主流市场永远不会看上有新功能的替代品。不相信小众产品有望瓜分、替代主流市场产品,甚至创造新市场的实力。
5	创新的高风险和不稳定,我准备好了吗?	敢于面对不利局面,你的创新可能被人认为简单、低质、不需要、一文不值,只要明白自己的产品满足了新市场、新需求,承认缺点,勇往直前。	破坏性创新具有的高风险不值得尝试,回到模仿大公司的老路上。以己之短攻彼之长。

倚动实验室简介

2010年创办，博士生导师领衔，本、硕、博协作，融学术研究于社会实践，以创新促进创业，培养大学生“互联网+”创新创业能力。

从0到1，开创多个初创项目，成长良好。承担了一系列研究项目。

研发多门“互联网+”双创课程，出版10部“互联网+”双创系列教材，被全国100多所高校选用。

开发多款App，“卓导App”服务于考研大学生与研究生导师，“豹考通App”服务高考考生与家长，力图实现优质资源最佳匹配。

- ◆ 清华大学出版社手机软件与移动商务系列教材总主编
- ◆ 教育部Google支持的产学研合作协同育人项目课题组
- ◆ 国家自然科学基金、全国教育科学规划项目课题组
- ◆ 全国“互联网+”创新创业教育协作联盟服务团队
- ◆ 大学生手机软件设计竞赛技术支持骨干团队
- ◆ 江西省科技型中小企业技术创新基金项目课题组
- ◆ “移动互联网+”专业方向校企合作、研究生联合培养基地
 - ① 软件工程(移动商务方向) 江西财经大学软件与通信工程学院
 - ② 教育技术学(移动学习与手机软件开发)，江西财经大学、江西农业大学等
 - ③ 计算机应用技术(手机软件班) 江西机电职业技术学院

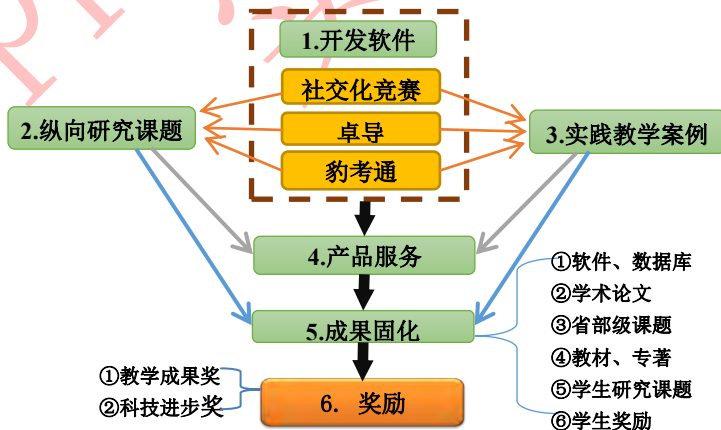
1. 定位、文化和培养目标

- ◇ 定位：“互联网+”双创训练营
- ◇ 文化理念：不拼爹，创先机
- ◇ 培养目标：“互联网+”公司合伙人

2. 产品领域、训练方法与运营模式

- ◇ 产品领域：O2O服务
- ◇ 训练方法：精益创业
- ◇ 运营模式：市场化驱动

3. 实践教学与学术研究相结合路径图



全书围绕一个真实项目，通过一个个小案例，引导读者短时间内熟悉一个综合 App 的开发，以培养 App 程序员的独立开发能力。

全书包括 App 项目体验、需求分析、App 全栈开发微案例、无对话框的单页界面设计、含对话框的单页界面设计、多页面切换效果设计、绘制趋势图、使用工具管理 SQLite 数据库、SQLite 数据库的操作模块设计、使用 MySQL 管理网络数据库、App 网络服务器端设计、App 客户端与服务器交互设计、App 集成测试和 App 全栈工程师创业思维训练等内容。

App

QUANZHAN GONGCHENGSHI
SHIZHAN ANLI JIAOCHENG

策划编辑 仲玉山
责任编辑 刘小连
仲玉山
封面设计 邓家珏

ISBN 978-7-5493-5914-1



9 787549 359141 >

定价：78.90元