1. 软件开发参考代码或链接或资料

[Android studio 自带虚拟机联网 - 哔哩哔哩](https://www.bilibili.com/opus/781505987323363349)

[Android中的Lottie动画入门 - 简书](https://www.jianshu.com/p/ddc22b094bd6)

[Android开发实战：利用系统自带日历控件实现高效日期管理 - 云原生实践](https://www.oryoy.com/news/android-kai-fa-shi-zhan-li-yong-xi-tong-zi-dai-ri-li-kong-jian-shi-xian-gao-xiao-ri-qi-guan-li.html)

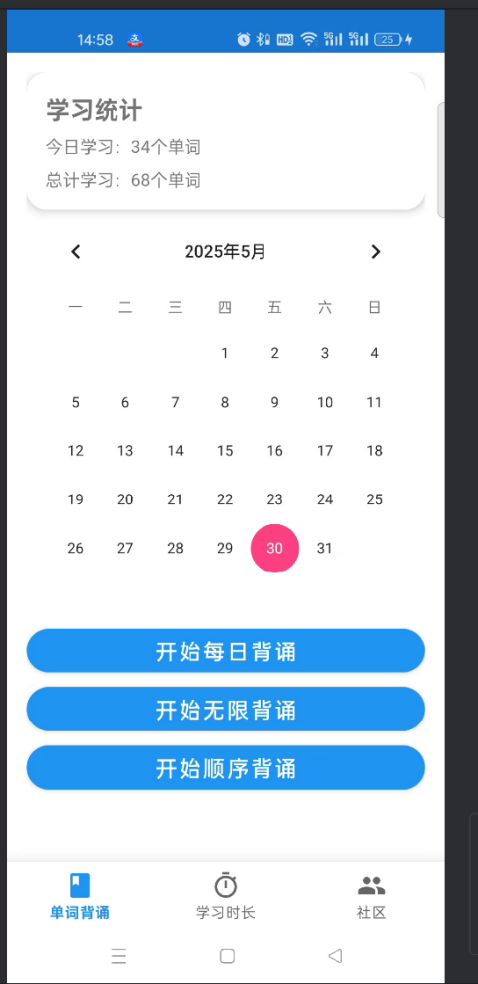
[Android：Fragment详细介绍 & 使用方法解析\_android fragment-CSDN博客](https://blog.csdn.net/JMW1407/article/details/114069742)

[Android计时常用的7种方式\_android 计时-CSDN博客](https://blog.csdn.net/cpcpcp123/article/details/88545040)

[OkHttp网络请求框架在Android中的使用\_android 网络请求框架-CSDN博客](https://blog.csdn.net/IH_LZH/article/details/139087236)

1. 本软件主要功能或新增创新点介绍（包含本人开发部分的介绍）
2. 功能一：单词背诵

用户在完成相关权限的授权后，进入到app中，通过底边的导航栏可以跳转功能模块，点击单词背诵后可以跳转到单词背诵模块：



页面对用户的背诵记录进行展示，可以通过点击日历切换日期展示不同日期的背诵情况，同时统计了总计背诵的单词数量。

点击每日背诵后，app会抽取10个用户尚未完成背诵的单词挨个展示英文原文，用户在点击遮挡板后会显示词义，通过点击三个记忆按钮“记得”“忘记”“没见过”对用户对该单词的记忆情况进行更新。

无限背诵：在每日背诵的基础上不设置上限，会不停的抽取单词背诵

顺序背诵：在以上基础上，会根据单词在考研英语的出现频率顺序抽取单词展示

1. 功能二：学习时长记录

用户通过导航栏跳转到学习时长记录界面：



用户可以通过点击模式切换切换倒计时模式以及计时模式来切换计时模式，在完成倒计时或者用户自主结束计时后还会对学习时长做记录，并统计总学习时长展示。

1. 功能三：社区登录同步数据

用户通过导航栏跳转到社区模块，在保证服务器连接成功的情况下，完成登录验证后，可以将数据上传至服务器中，同时也可以将服务器数据同步到本地app上。

1. 突出创新点：

使用本地sqlite存储用户学习记录保证离线使用，使用service实现对计时功能的管理，通过后端服务器部署实现用户多台设备的数据同步，使用了lottie动画效果展示

5）在分组中，本人完成了本地sqlite部署以及存储数据结构设计，三个模块的全部基础开发，完成了后端的基础配置部署工作。

1. 实验环境配置说明

1）代码运行需要的android库：  
 implementation("androidx.core:core-ktx:1.10.1")

implementation("androidx.appcompat:appcompat:1.6.1")

implementation("com.google.android.material:material:1.9.0")

implementation("androidx.constraintlayout:constraintlayout:2.1.4")

implementation("com.airbnb.android:lottie:3.4.2")

implementation("com.squareup.okhttp3:okhttp:4.12.0")

testImplementation("junit:junit:4.13.2")

androidTestImplementation("androidx.test.ext:junit:1.1.5")

androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

implementation("com.google.code.gson:gson:2.10.1")

implementation("com.squareup.okhttp3:okhttp:4.12.0")

implementation("com.fasterxml.jackson.core:jackson-core:2.15.2")

implementation("com.fasterxml.jackson.core:jackson-annotations:2.15.2")

implementation("com.fasterxml.jackson.core:jackson-databind:2.15.2")

完成包的导入运行即可，sqlite的模版db文件已在assets文件中存储，如apk未能正确导入db文件也会使用模版db文件复制一份文件。

2）后端配置：  
 运行框架：SpringBoot Version：3.1.2 （jdk：17）

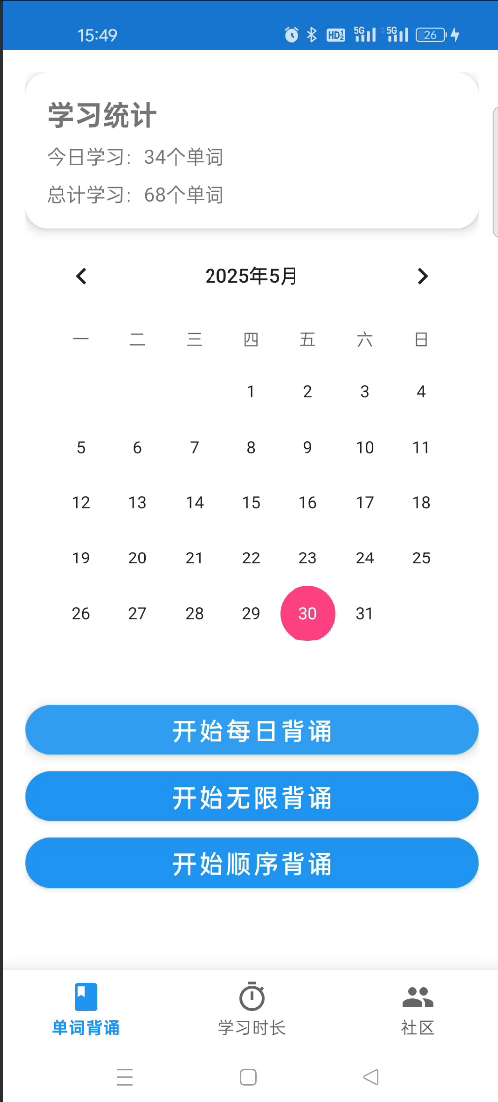
数据库：mysql （需要提前创建好表结构）

依赖项：mybatisplus，hutool，knife4j swagger（用于测试）等

4. 功能验证：（软件操作说明+运行截图，避免大段代码展示）

1）功能一： 单词背诵

首先进入app完成授权，可以看到app的单词背诵欢迎界面：



接下来检查背诵效果：  
 点击每日背诵查看效果：



这里会在sqlite数据库中查询10个状态为尚未背诵过的单词并封装提取到Fragment中，并对获取到的数据解析set到layout的视图中。

可以看到进入后存在遮挡板遮挡词义，在用户点击后会显示词义，并提供选项；



可以看到app提供了三个选项给用户，用以选择以更新用户对单词的记忆情况，当用户点击“没见过”“忘记”后，还会将该单词移至背诵队列的最后（实际实现是使用了另一个list存储需要复习的单词），并在用户完成其他单词的记忆后再次背诵：



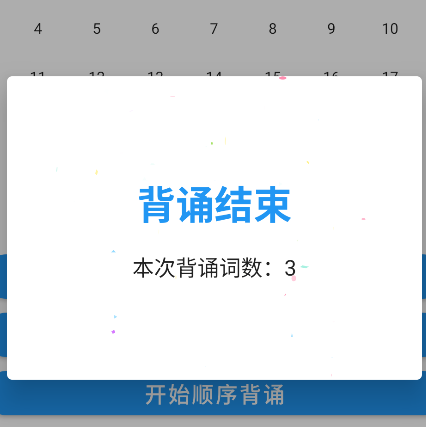
完成后会跳出lottie动画效果弹窗庆祝：



现在检查无限背诵：



可以看到展示效果和每日背诵差不多，同样是遮挡板遮挡词义，点击后显示，这里额外提供了一个“结束背诵”的按钮来结束无限背诵：



现在检查顺序背诵：



可以看到顺序背诵在前两者的基础上还提供了“上一个”“下一个”的选项，用户在点击按钮会跳转到对应序号的单词位置并显示。同样的，点击“结束背诵”后会统计本次背诵单词（点击“记得”选项的才会被记录）：



2）功能二：学习时长记录（专注模式）

用户点击导航栏“学习时长”后，跳转到专注模式模块，可以点击切换模式来切换计时的方式：

“倒计时模式”：在用户设置好计时时间后，点击开始计时，会开始倒计时，在时间结束或用户自主结束后停止计时并展示专注时间，统计到总时长中：







“计时模式”：用户点击开始计时立刻开始计时，同样的，用户点击结束计时后会停止计时并展示专注时间，统计到总时长中：

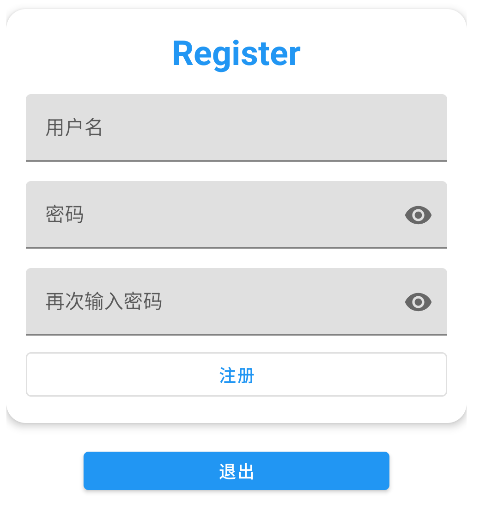


3）功能三：社区数据同步

用户点击导航栏“社区”模块后跳转到社区模块，如在本地sharedPreferences中未能找到userid，则会进入登录页面；如已有userid，则会直接跳转到操作界面：



登录页面可以看到form登录框，如无登录账号，则需要注册，可以点击“注册”按钮跳转到注册界面：



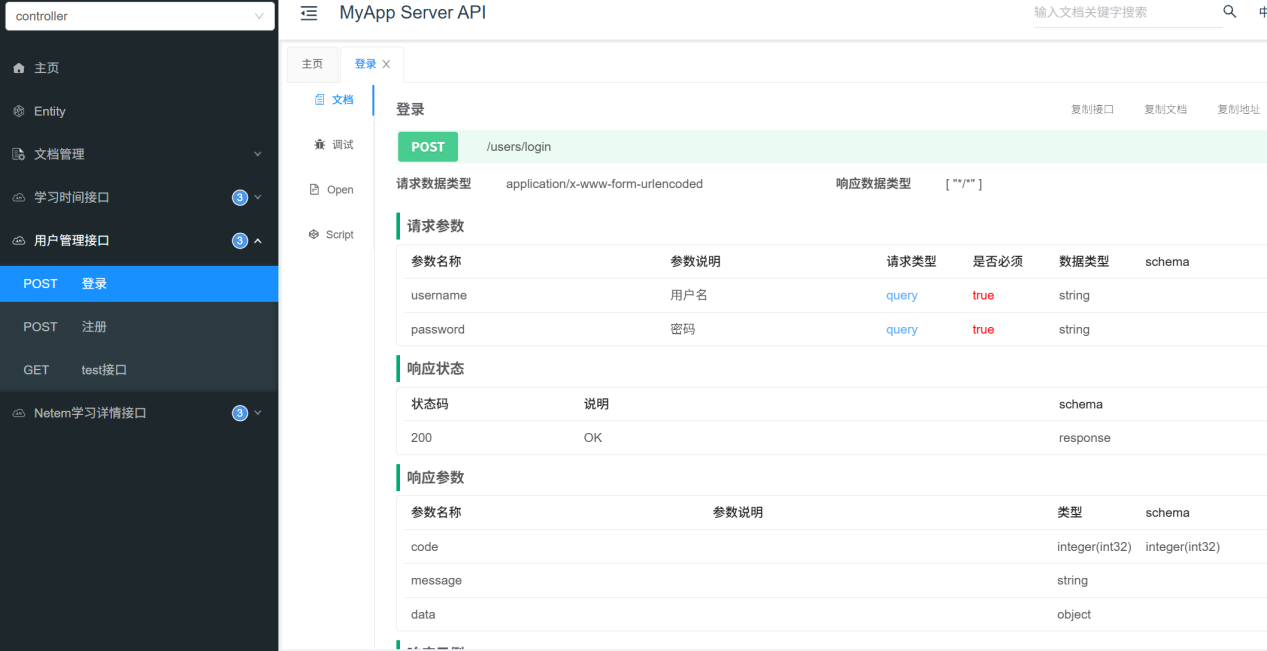
当然这一切的前提是服务器完成了部署且开启了对应端口，在前端，这里提供了一个“设置服务器信息”的按钮对服务器信息做配置，设置完成的信息会被存储到sharedPreferences中，后续登录注册等操作会直接从中提取：

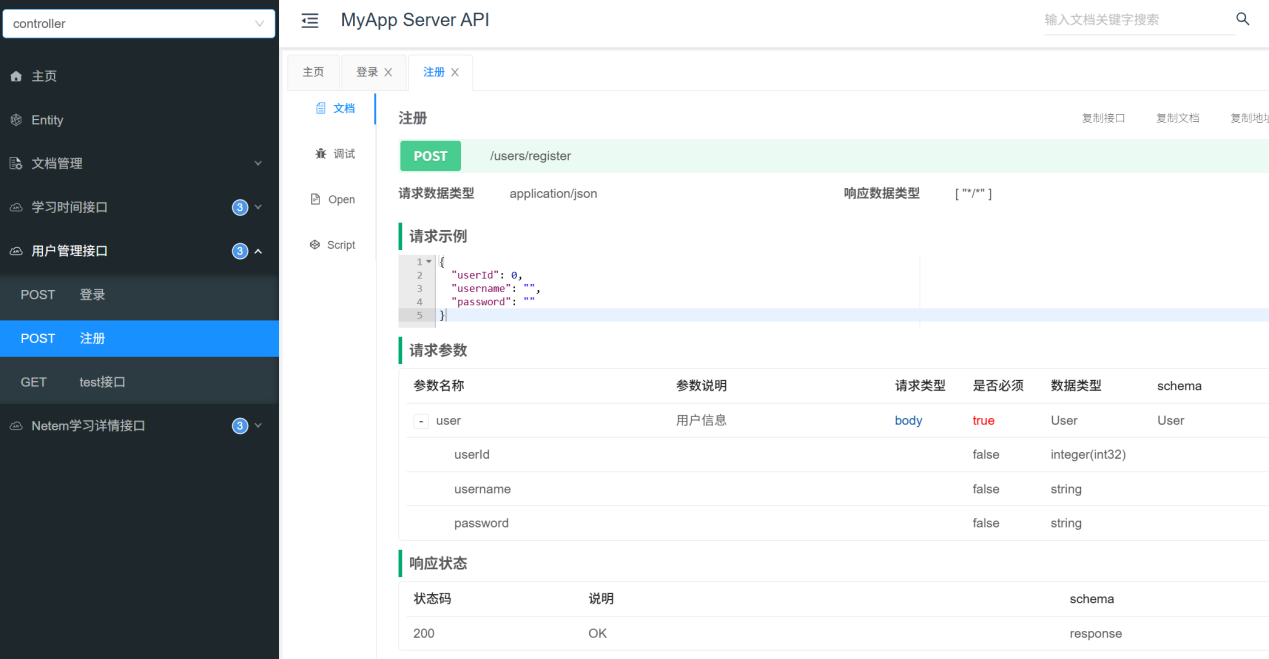


在完成了后端的部署后，我们就可以进行登录等操作了：



登录通信实现方式是okhttp发送request请求，这里简单展示一下账号模块的后端的接口（对后端接口的详细说明在app项目文件夹下controller.md中给出），后端项目使用了knif4j swagger接口测试文档，项目启动后可在[MyApp Server API](http://localhost:8080/doc.html" \l "/home)下查看详细接口并调试：





详细接口见“接口api说明.md”文件

4 ) 突出创新点：

使用本地sqlite存储用户学习记录保证离线使用，使用service实现对计时功能的管理,通过后端服务器部署实现用户多台设备的数据同步，使用了lottie动画效果展示

5. 个人在开发过程中遇到的问题及未来还需要增加的功能

遇到的问题以及解决方式：

初始化数据库存在问题：由于需要外界导入db文件（单词表），我们需要一个不会被压缩破坏的方式存储文件，后续了解到可以将文件放入assests文件夹来保存原原本本完完整整的文件，随后只需要在sqlconnection中编写getwritabledb等方法中加入检查，如无则import assests的db文件即可解决问题，这样就保证了打包apk不会出现db文件加载失败的问题。

项目选择的构建方式是mainActivity+多Fragment，一开始的设计没有实现对多Fragment的维护，选择了点击导航栏按钮后就会replace当前的Activity管理唯一一个fragment来实现切换，这就会导致一个情况：切换Fragment后，原来的fragment的状态就会被销毁，像是计时功能以及背诵背到一半就会被迫终止。最后通过各自init了一个Fragment并维护各自在mainactivity的layout中的container来实现，在点击导航栏切换模块时设置各自container的visibility来实现模块功能的不中断切换。

实际部署到自用手机上存在service无法正常运行，后端挂起运行停止导致计时结束等问题，但由于无法查看报错导致问题无从查起。最后通过手机usb连接到android studio使用开发者模式调查logcat，发现是权限问题，后来在welcome界面会对后端运行权限以及通知notification权限进行检查，如没有则会提示用户未授权，并intent跳转到对应权限设置页面协助用户设置。

虚拟机联网存在问题，由于android studio给虚拟机设置的ip一般和本地的电脑的ip网段不同，这就会导致联不通网，就连本地的端口也连接不通，从而无法连接到本地部署的后端服务器，这个问题可以通过两种方式解决：1.通过自己的手机usb连接到studio并连接电脑的热点，用实体机来连接后端服务器；2.通过更改虚拟机启动设置，我们可以找到studio的启动位置，设置启动指令：emulator.exe -avd Medium\_Phone\_API\_35 -dns-server 114.114.114.114 来修改虚拟机的DNS实现网络联通来实现对服务器的连接，具体实现[Android studio 自带虚拟机联网 - 哔哩哔哩](https://www.bilibili.com/opus/781505987323363349)（不是专栏的，是底下的评论的方案）



需要增加的功能：

服务器的同步数据功能还未完成，后续需要更新完成。

自动登录的验证太过简单，需要设置JWT令牌实现登录验证，后续可考虑加入Redis检查自动登录令牌。

目前对单词的背诵方式只是简单的随机抽取且记忆的程度没有做详细的分析，后续还需要结合单词的重要性，记忆忘记曲线等多方面对单词背诵策略作更新。

日历的实现比较简单，后续还可以加入签到，连续打卡等功能。

Service的实现与Fragment虽然解耦了，但是两个计时的runable会更加占用内存，可以考虑viewmodel的方式共享数据，在关键点对Fragment作更新维护即可实现化简。