# 联系方式

手机： (+86) 13902540358

Email： [ryanyans32@163.com](mailto:ryanyans32@163.com)

QQ/微信： 421078521

# 个人信息

袁柏宁 / 男 / 26

本科 / 广东石油化工学院 / 计算机科学与技术

籍贯：广东 · 东莞

期望职位：Android开发工程师

# 技能清单

**自我评价：** 一、扎实的Android开发经验，主要在云存储、互联网金融行业工作，也涉及过社交等其他类型的项目；

二、熟悉互联网移动项目开发及发布流程，具备独立开发项目模块的能力；

谢绝外包项目及人力外派性质岗位招聘！

以下列举了一些基本技能，如有缘分，期待详聊。

1、 熟练掌握 Java 编程，具有扎实的技术功底，有良好的编程习惯；

2、 熟练掌握 Android Studio/IDEA开发环境及Git/SVN版本控制工具使用；

3、 熟练掌握 自定义控件（https://tower.im/teams/603160/todos/322/）、动画、布局的设计开发，对系统启动流程（https://upload-images.jianshu.io/upload\_images/1858589-a07eaf45e6552da4.png）、绘制流程源码有一定研究；

（https://img-blog.csdn.net/20160922170541057）

4、 熟悉 Socket、TCP/IP、Http等通信协议，对OkHttp源码有一定研究（四大核心类：OkHttpClient、Request、Call 和 Response; 请求队列、连接池、拦截器（AppicationIntercept、RetryAndFllowIntercepter、BridgeIntercepter、CacheInterIntercepter、ConnectIntercepter、NetworkIntercepter、Call、RealCall、Dispatcher））；

5、 掌握 Stetho+OkHttp+Chrome及其移动端网络调试与分析经验；

6、 熟悉 WebView、JS-Native交互、HTML5混合开发；

（最核心的就是URI的拦截和MessageQueue消息队列，前端跟本地都会维护一个MsgQueue队列、点击先进去flushQueue刷新队列重新loadUrl跟前端通信并设置一个Callback回调，前端把消息队列内容封装起来集合回传给WebViewClient，通过浏览器内核ShouldOverrideUrlLoading中，最终进入我们的callBack上，然后拿到拦截的Uri跟前端的CallBack然后本地处理，Native调JS同理）

1. 熟悉 MVP架构、组件化、模块化、插件化开发技术，及在项目中的应用经验；

（纵向：上层业务层组件、公共业务组件层、基础服务组件层，上层依赖下层组件）、模块化（横向：申购模块、登陆注册模块、消息中心、支付模块、团宝箱等等的模块，不做模块化分离非常难维护）、插件化（核心的思想就是将复杂的宿主App拆分成独立的插件App，然后去动态替换，像插件化技术常见的应用就是热修复比如Tinker等等的这种热修复技术框架，本质上就是动态获取Android虚拟机的PathClassLoader，找到他的dexPathList类中的dexElements数组，这是个有序数组并且存放的是类加载的信息，然后将插件App的类动态注入数组前面，实现一种动态替换效果- 生成补丁包（DexDiff算法、资源ResDiff-MD5比较-Png优化问题）-合成补丁包（逆向DexDiff算法）-加载补丁包（类文件Dex、Path、BootCalssLoad、PathList、DexElements、资源Hook Assest 重新创建Assest传入补丁Path路径 反射替换AssetManager） ...dexDiff、逆向Diff算法、类、资源ResDiff-MD5比较-Png优化问题、Dex、Path、BootCalssLoad、PathList、DexElements数组、、插件的话还涉及Activity启动流程(AMS)的Hook与资源获取Assest类的Hook流程）

8、 熟悉 性能优化，有自己的一套性能优化方案；

9、 熟悉 企业级代码质量规范管理机制及代码QA检测工具；（Android Lint静态代码检查工具）

其实自己团队也是在一直完善代码规范上的问题，像之前我们在Wiki制定了完善的规范要求、内部也推动过一些本地静态代码扫描的工具像CheckStyle、PMD。。并加强代码的CodeReview。最终实施下来发现最苦的就是合并代码的审查同事，其实团队内也发现这个问题，也进行过讨论，最后选择AndroidLint并自定义Lint的本地规则结合自动化构建，集成进去跑生成代码规范结果、一旦有高优先级的代码规范问题就会直接构建失败并邮件、钉钉通知到个人；

掌握 Gradle构建优化（核心就是变体，不同环境构建不同的变体来依赖不同的Plugin插件，比如我们开发都会涉及到很多环境，像测试、灰度、正式、本地调试，他们依赖的插件其实都不同、还有特定的变体包如Performance性能测试包等等依赖的Task都是不一样的，那么我们本地调试就可以建立一个最轻量级的fastDev变体，打造最轻量级的构建环境，把不相关的构建插件删掉、静态资源的检测、混淆等等的这些task配置都停掉； 不同的环境不同的变体构建不同的Gradle脚本）

，具备一定自动化构建、持续集成(CI/CD)经验；（流水线超时时间 JenkinsFile、Pipeline、跑代码扫描、性能监控的Job ）

**[CI最重要最核心要做的就是监控-监控-监控、监控代码、监控APK包、监控启动时间、构建时间、性能监控]**

1. 熟悉 Devops平台开发；

（自己中心内部孵化的项目，主要针对团队内开发流程的持续优化，改善开发过程中低效率的地方。需求 ->设计 ->开发 ->测试 ->监控与维护）https://blog.51cto.com/lookingdream/2447988?source=drt

# 工作经验

2019/5 - 至今 **Android开发** | 前端研发部

广东一一五科技股份有限公司

云/存储/社交 | 民营公司

**工作描述：** 1、公共组件、功能类开发（Java & Kotlin），核心业务开发、迭代与维护；

2、性能分析、优化、Js-Native封装与维护；

3、本地Gradle构建优化研究与实施；

Gradle Plugin：<https://blog.csdn.net/huachao1001/article/details/51810328>

【Android】函数插桩（Gradle + ASM）：https://www.jianshu.com/p/16ed4d233fd1

4、Aspectj + IntelliJ IDEA 插件开发； TD-571、https://github.com/alfredxl/TuandaiBiPlugin

[Android中的AOP的实现及AspectJ的使用](https://blog.csdn.net/leol_2/article/details/105416332)

5、协助Devops平台搭建、Jenkins平台的维护；

6、性能分析与优化（基于Tecent-GT）；

7、Bugly日常维护、处理，及对接跟踪线上Trance文件；

2017/12 - 2019/4 **Android开发** | 移动项目部

东莞团贷网互联网科技服务有限公司

金融/投资/证券 | 民营公司

**工作描述：** 1、业务层模块的开发迭代工作，如签到，申购，标的列表、底标详情等业务；

1. 旧模块旧功能的维护、Bugly日常跟踪与处理；
2. 整体项目优化；
3. Wiki文档编写与整理；

# 项目经验

2019/5 - 至今 **115网盘 / 115TV / 115组织**

所属公司： 广东一一五科技股份有限公司

项目描述： 115科技是国内最早一批从事云存储研发的企业，115网盘、115组织作为自主研发的两大核心产品，涵盖个人数据存储、管理、生活社交、视频影音等产品服务，其中115网盘已拥有百万级别的日活量。

Android端采用MVP架构，采用组件化、模块化开发模式对业务进行抽离，同时正逐步对单一业务组件进行Kotlin重构，以及通过持续集成手段对应用末端进行有效的把控，保证应用持续高效稳定的运行。

**责任描述：** 1、公共组件、功能类开发（Java & Kotlin），业务模块的开发迭代与维护，如文件、备份、记录、社区、社交、VIP等等；

1. 组件层的Kotlin重构（整合重复代码的逻辑、拆分复杂业务逻辑方法、通过模板方法、外观等设计模式整理对外的api、补充相关注释）、自定义控件的开发维护、与各领域(产品，交互，接口、测试) 的沟通日常；

·Kotlin协程优势在于线程切换、异步管理-简洁清晰-替代RxJava。

·默认参数-有效避免方法的重载、也能替代Java中的Builder模式，减少大幅代码量。

·其他的像空判断、data class、扩展函数、字符模板。

缺点：反射支持不够好、有时拿不到具体类信息导致报错，构建时间普遍会比较长-同比纯Java的模块编译时长高2倍。

3、基于JsBridge对本地Js-Native交互逻辑进行重构与封装；核心理解-13 https://www.jianshu.com/p/bdf78331fea1

4、本地Gradle构建优化研究与实施，本地编译速度提高20%；TD-546~555

Gradle配置优化：持续升级Gradle版本、开启守护进程、开启并行编译、提高编译时的JVM内存大小、提高开发变体的minSdkVersion到21或24 (高版本AS已弃用、新版本AS自动上传API版本匹配)、还有优化注解处理器、选择支持增量注解处理器的版本。

很多官方或三方提速工具InstantRun、Freeline等：由于项目内存在其他编译期的AOP代码注入的插件、导致这些工具不可用。

Gradle变体构建：fastDev开发专用变体--删减部分非业务相关的task、避免执行非必要的task比如一些静态代码检测、资源检测等等Task、核心开发变体上构建不集成语言国际化、只保留默认语言、只保留固定dpi的图片资源、优化lintOption静态代码检测-停用部分检测配置、优化aaptOptions图片格式检测、取消ShrinkRes严苛资源扫描、取消fastDev变体的混淆。

整体项目优化：重复类资源整合、无用类资源删除、WebP图片的压缩、避免动态版本的依赖...

5、Aspectj + IntelliJ IDEA 插件开发，埋点开发效率提高50%以上；TD-571、https://github.com/alfredxl/TuandaiBiPlugin

[Android中的AOP的实现及AspectJ的使用](https://blog.csdn.net/leol_2/article/details/105416332)。

埋点插件核心：

AspectJ（aspect切面-Aspect类名、JPoint连接点-捕获的方法、PointCut切点-aspect方法/匹配（@Around("execution()）、Advance具体增强实现-注入类型Before、After、Arrond）。

AspectJ是通过对目标工程的.class文件进行代码注入的方式将通知(Advise)插入到目标代码中。

第一步：根据pointCut切点规则匹配的joinPoint；

第二步：将Advise插入到目标JoinPoint中。

IDEA插件开发：PsiUtilBase获取选取的文字、类名、方法名；再通过LocalFileSystem、VirtualFile、JavaDirectoryService对象创建文件夹与文件-Bi注解类、Aspcet类；通过JavaPsiFacade.getElementFactory()去获取当前元素PsiClass.getModifierList()获取需要写入的类、PsiMethod、PsiElement、PsiDirectory、PsiJavaFile等等文件系统类去控制文件结构的输出。

====IDEA插件使用VirtualFile、JavaDirectoryService生成类信息；APT 注解处理器一般用Filer与JavaPoet这一套API协助生成类文件====

1. APK瘦身，国内渠道缩减26.4%、GooglePlay渠道缩减19.6%；（核心：MD5文件去重-重复类-资源删除、 无用类-资源删除、三方组件等、 WebP图片格式压缩、 So库配置、低频架构删除保留：开始只保留V8a、实测线上还有有兼容问题、最终方案：Google渠道同时保留V7a与V8a，国内渠道只保留V7a架构； 其他：Shape代码就不用图片、Json动画替换帧动画图片、能用.9图就不用png、Assets文件字体文件等等只提取用部分、AndResGuard资源混淆-（资源ID与资源全路径的映射关系都是记录在arsc文件上、混淆减少众多文件名）、压缩APK包上的resource.arse映射文件）--工具ApkChecker

国内外渠道资源做区分，国内渠道保留v7a架构so库、国外渠道同时保留v7a及v8a。多语言国际化国外渠道保留、国内渠道只保留默认语言

APK包大小检测也加入持续基础构建上，每次提交时都会触发前置Lint静态代码检测、并且在构建后生成APKCheker检测报告等等。

**7、性能分析优化（基于Tecent-GT），核心页面启动及布局构建时间缩短20~30%；（核心理解-16）**

* + 1. 分析工具原理：启动耗时、方法耗时、View构建耗时、View绘制耗时、流畅度FPS （反射+代理Hook、AOP、Chpf）
    2. 然后做优化、优化前理解启动跟布局构建、绘制的整体流程，整理可能存在的耗时地方、或值得优化的地方。
    3. 性能优化：布局层级、标签、避免RLLL层级嵌套、避免BackGround、避免使用Wrap\_Content、Spannable替代textView处理富文本、自定义View的话避免多次绘制并设置cavans.drawRect(x,x,x,x)绘制区域、尽量不使用帧动画、可用Json动画或属性动画替代帧动画、引入X2J架构、编译器动态将XML布局转Java方式构建。
    4. 启动优化：对于Application-onCreate，避免执行一些数据IO跟初始化动作，对于业务相关的SDK就懒加载、使用时再进行初始化，对于优先级比较低的SDK就延时初始化或放到IdleHandler上空闲队列上去执行，将一些本地时间修正的请求跟校验工作放到异步执行。然后对闪屏页的Logo图片进行WebP格式的压缩，等到页面绘制结束后回调再去执行隐私政策弹窗跟BI弹窗的校验与显示逻辑。

1. 加入中心Devops团队，协助维护Jenkins自动化构建、持续集成，代码扫描，性能监控、Jenkins Job及PipeLine等

Merge - Checkout - 静态代码检测报告 - APK特定渠道打包 - ApkCheck包检测报告 - Appium+STF云台跑自动化测试场景、性能检测分析报告 - Release版本加固 / 重签名...一系列流程来保证版本构建的质量。

定时任务、Gitlab-Hook自动触发构建、前置代码检测生成报告-超阈值邮件警告、自动化构建-不同渠道变体、APKChecker进行包大小检测并生成报告、特定Performance变体包进行自动化测试、跑性能测试脚本生成性能测试报告、正式包还会触发加固重签等。

Jenkins：Agent、Options（配置超时时间timeout、重试次数retry）、Stages（不同的构建节点）、Stage、Step、Post（构建完成、失败、被终止的处理）<https://blog.51cto.com/lookingdream/2447988?source=drt>

<https://gitbook.cn/books/5e42ab341cd0a1288a873070/index.html>

（Pipeline: agent、option-超时/重试次数、enviroment-变量、parallel-并发、trigger-每天/触发器、post-失败/成功必操作、Stages-stage-step）

9、Wiki编写、技术分享、Bugly维护、线上Trances文件跟踪；

(App版本Crash率由19年底的千分之三降至21年初的万分之八)

NullPointException、IndexOutOfBoundsException、IllegalArguementException（非法参数）、ClassCastException。

1.x 部分集合类Key、Value值不能为空。（HashTable、ConcurrentHashMap）

2、注意主线程、子线程的操作，像自定义控件、或者对控件绘制相关的方法必须放主线程。

3、先定位具体的类、然后定位具体的方法或View动作，如果页面是复杂的布局层级的话、还能对比具体的View ID来定位，对比APK包上面的资源ID - 根据包上面的resources.arsc资源索引文件来定位。

4、Bugly：有些比较难复现的问题比如资源、销毁等等，可以尝试开启系统不保留活动。

5、Trace - ANR的问题、不能只盯着看主线程，主线程基本看不出问题、必须去其他子线程上定位问题、先大概过滤下上层业务类信息堆栈再尝试定位。

（这一两年来慢慢对开发流程上规范、代码层面上的规范、性能层面上的规范慢慢会对这些做出要求、通过自测、持续集成、线上监控等手段去协助我们更好地维持每个版本的稳定性）

2017/12 - 2019/4 **团贷网**

所属公司： 东莞团贷网互联网科技服务有限公司

项目描述： 团贷网作为前国内大型P2P平台，涉及的业务广泛，复杂，而移动端承担了绝大部分的客户流量，具有百万级别的用户量。

项目整体采用MVP设计思想搭建，组件化、模块化开发，并基于Retrofit(动态代理-简洁+能同时代理多个不同的类、不用新建子类、不影响现有类功能) + OkHttp + Glide + RxJava等成熟三方框架的二次封装，以尽大可能实现层级解耦，保持高效稳定的运行。

**责任描述：** 1、参与业务层模块的开发迭代工作，如签到、分享、申购、标的列表、底标详情等业务模块；

2、针对项目中旧模块旧功能的维护、Bugly日常跟踪与处理；

3、推动代码审查及代码规范检测制度落实；

4、整体项目优化；

5、Wiki文档编写与整理；

# 其他经历

通过CET-4考试，能独立阅读英文API文档，良好的英语理解能力；

大学期间加入院系创新实验室，曾多次参加校、省电子创新竞赛并获得优异成绩；

大三期间兼任实验室学生团队组长，管理7人团队，配合老师工作，协同团队成员任务；

# 关于我

保持对新技术好奇、技术钻研，喜欢在团队编程中找到成就感和集体荣誉感；

团队合作意识强，积极主动与伙伴沟通，乐于帮助他人，共享技能与经验；

( Logical Thinking && Testing ) === Best Software ；

# 致谢

感谢您花时间阅读我的简历，期待能有机会与您共事！

**监控终极整理：**

**开发阶段：**

1. **StrictMode：主线程耗时情况（方法耗时、数据库IO、网络IO）、内存泄漏（Activity泄漏、Sqlite泄漏、实例泄漏）**
2. **SysTrace、Perfetto：启动流程完整时序图、周期图，分析启动流程关键节点的耗时情况。**

**CI集成阶段：**

1. **通过反射+动态代理Hook启动流程关键节点、或者通过AOP方式在编译器动态注入统计代码，来获取启动流程上的耗时。**
2. **通过设置Looper的Printer来监控ANR、或启动一个Thread不断定时发送Msg并睡眠间隔时间-醒来检测Msg消费情况。**
3. **通过LayoutInflateCompt.setFactory2（callback-onCreateView），监听单个View的构建耗时。**

**优化方案：**

**1、启动优化：比如支付、分享、一键登录等等的业务相关的SDK，没必要在App首次启动的时候就同步的进行初始化，还有优先级较低的SDK也可以延时或放入idleHandler等到空闲队列再执行、还有像一些。**