



移动测试 用户指南

文档版本: 20220421



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。 如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	⚠ 危险 重置操作将丢失用户配置数据。
⚠ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔〕) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	⑦ 说明您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.打开控制台页面	<mark>0</mark> 8
2.上传应用	10
3.上传符号表	11
4.管理测试用例	12
4.1. 打开用例管理页面	12
4.2. 组织层次说明	12
4.3. 管理脚本	13
4.3.1. 新建脚本目录	13
4.3.2. 新建/导入脚本	14
4.3.3. 录制脚本	16
4.3.4. 编写脚本	24
4.3.4.1. 编写Android测试脚本	24
4.3.4.2. 编写iOS测试脚本	27
4.4. 管理参数池	30
4.4.1. 新建参数分组	30
4.4.2. 新建/导入参数	31
4.5. 管理用例库	32
4.5.1. 新建用例库版本	32
4.5.2. 新建/导入模块	33
4.5.3. 新建/导入用例集	35
4.5.4. 新建用例	37
5.实施Android测试	42
5.1. 兼容测试	42
5.1.1. 提交测试	42
5.1.2. 查看概览信息	45
5.1.3. 查看测试报告	46

5.1.4. 测试报告说明	46
5.2. 功能测试	52
5.2.1. 提交测试	52
5.2.2. 查看概览信息	55
5.2.3. 查看测试报告	56
5.2.4. 测试报告说明	56
5.3. 性能测试	61
5.3.1. 管理测试用例	61
5.3.2. 提交测试	64
5.3.3. 查看测试报告	65
5.4. 远程调试	67
6.实施iOS测试	76
6.1. 兼容测试	76
6.1.1. 提交测试	76
6.1.2. 查看概览信息	79
6.1.3. 查看测试报告	80
6.1.4. 测试报告说明	80
6.2. 功能测试	85
6.2.1. 提交测试	85
6.2.2. 查看概览信息	88
6.2.3. 查看测试报告	89
6.2.4. 测试报告说明	89
6.3. 性能测试	93
6.3.1. 管理测试用例	93
6.3.2. 提交测试	96
6.3.3. 查看测试报告	96
6.4. 远程调试	98
7.管理测试任务	105

8.申请专家测试	106
9.咨询专有云服务	113
10.查看计费	114
11.授权RAM用户	115
12.日志解释与解决方案	117
12.1. 崩溃时错误	117
12.1.1. android_content_ActivityNotFoundException	117
12.1.2. android_content_res_Resources_NotFoundException	117
12.1.3. android_database_sqlite_DatabaseObjectNotClosedExcep	117
12.1.4. android_database_sqlite_SQLiteCantOpenDatabaseExcep	118
12.1.5. android_os_DeadObjectException	119
12.1.6. android_os_TransactionTooLargeException	119
12.1.7. com_android_internal_os_RuntimeInit_UncaughtHandler	121
12.1.8. java_io_EOFException	121
12.1.9. java_io_FileNotFoundException	122
12.1.10. java_io_IOException	123
12.1.11. java_lang_ArrayIndexOutOfBoundsException	124
12.1.12. java_lang_ArrayStoreException	125
12.1.13. java_lang_ClassCastException	125
12.1.14. java_lang_ClassNotFoundException	125
12.1.15. java_lang_IllegalArgumentException	126
12.1.16. java_lang_IllegalStateException	128
12.1.17. java_lang_NoClassDefFoundError	129
12.1.18. java_lang_NoSuchMethodException	129
12.1.19. java_lang_NullPointerException	130
12.1.20. java_lang_NumberFormatException	131
12.1.21. java_lang_OutOfMemoryError	131
12.1.22. java_lang_reflect_InvocationTargetException	131

12.1.23. java_lang_RuntimeException 1	132
12.1.24. java_lang_SecurityException 1	134
12.1.25. java_lang_UnsatisfiedLinkError 1	134
12.1.26. java_lang_UnsupportedOperationException 1	138
12.1.27. java_net_MalformedURLException	139
12.1.28. java_net_SocketException 1	139
12.1.29. java_net_SocketTimeoutException	141
12.1.30. org_json_JSONException 1	142
12.2. 安装时错误	142
12.2.1. FAILED_OLDER_SDK 1	142
12.2.2. INSTALL_FAILED_CANCELLED_BY_USER	143
12.2.3. INSTALL_FAILED_CONTAINER_ERROR 1	143
12.2.4. INSTALL_FAILED_DEXOPT 1	143
12.2.5. INSTALL_FAILED_MEDIA_UNAVAILABLE	143
12.2.6. INSTALL_FAILED_NO_MATCHING_ABIS 1	143
12.2.7. INSTALL_FAILED_UID_CHANGED 1	44
12.2.8. INSTALL_FAILED_VERIFICATION_TIMEOUT	44
12.2.9. INSTALL_FAILED_VERSION_DOWNGRADE1	44
12.2.10. INSTALL_PARSE_FAILED_INCONSISTENT_CERTIFICATES	44
12.2.11. INSTALL_PARSE_FAILED_NO_CERTIFICATES	145
12.3. 启动时错误	145
12.4. Native_crash 1	145

```
1.打开控制台页面
前提条件
• 已注册/登录阿里云账户。具体操作参见: 阿里云账号注册流程。
● 已开通移动测试服务。具体操作参见:移动研发平台 EMAS > 快速入门。
• 已打开阿里云控制台首页。
操作步骤
1. 在阿里云控制台首页,将鼠标移动至左上角,单击 📃 ,打开快捷菜单。
2. 选择产品与服务 > 开发与运维 > 移动研发平台 EMAS,进入移动研发平台EMAS控制台。
    ? 说明
    您也可以在搜索框中输入移动研发平台EMAS,确认后进入移动研发平台EMAS控制台。
       Q 搜索.... 费用 工单 ICP 备案 企业
   产品与服务
                          〇 清输入关键词
    N 云服务器 ECS
                          最近访问
    🐨 云数据库 RDS 版
                          移动研发平台 EMAS
      专有网络 VPC
    ക
    ♪ 対象存储 OSS
                                                                   开发与运维
                                              大数据
                         弹性计算
    👃 负载均衡
                          云服务器
                                              大数据计算与分析
                                                                    开发与运维
                          云服务器 ECS
                                              云盾生大数据计算服务 MaxCompute
                                                                   移动研发平台 EMAS
    🗙 CDN
                          轻量应用服务器
                                               实时数合 Hologres
                                                                    应用实时监控服务 ARMS
    大数据开发治理平台 DataWorks
                         弹性加速计算实例
                                                                    Prometheus 监控服务
                                               Elasticsearch
    ● 域名
                         弹性云手机
                                         🕁 实时计算 Flink 版
                                                                    天速搭 CADT
    👶 云市场
                                               日志服务 SLS
                         计算巢服务
                                                                    云监控
                          VMware 服务 公测中
                                               智能开放搜索 OpenSearch
                                                                    云网管
                         无影云电脑
                                               DataQuotient 画像分析
                                                                    Web 应用托管服务 (Web+)
                                               图计算服务 Graph Compute
                                                                    Node.js 性能平台
                          无影云卓面
3. 在EMAS控制台中,单击产品列表区域的查看更多,进入产品介绍页面。
       (-) 阿里云 🛛 🌣 工作台
                                                         ● 【有奖调研】移动研发平台EMAS产品使用调研,动动手指,赢取精美礼品! 点击进入
                                                                                     查看更多>
                        研发工具
                                                          Mobile DevOps
面向多端应用场景,全生命周期的一站式研发支撑平台,帮助企业实现交付流程
化、目动化、数字化。
                         低代码平台Mobi
                         10070月十日141001
低代码开发平台Mobi:一站式可视化的应用研发平台,提供应用前后端的搭建。
托管、运缮服务,适用于Web、H5和小程序等应用。
                        平台服务
                         HTTPDNS
                                                          小程序Serverless
                         HTTPDNS是面向移动App的域名解析服务,具有域名防劫持、精准调度、实时解析生效的特性。
                                                          小程序云是词里云面向小程序场景提供的一站式云服务,帮助开发者实现一云多
端的业务战略。
4. 在产品介绍中选择移动测试,单击立即使用,即可进入移动测试控制台。
```



2.上传应用

前提条件

已打开移动测试控制台页面。具体操作参见:<mark>打开控制台页面</mark>。

操作步骤

1:在控制台页面,打开**测试管理 >应用管理**。

2:在应用管理页面,单击上传应用按钮,打开上传应用右侧栏。

移动研发平台EMAS / 移动测试 / 应用管理

移动测试

测试产品	Ň	◆ 上传应用	3 搜索应用包名称					
测试管理		应用包名称	终端类型	版本	文件大小	包名	上传时间	操作
用例管	理	爱奇艺	i Android	V11.10.0(800111050)	48.96M	com.qiyi.video	2020-11-17 20:26:56	下载 删除
应用管	····	testme	📥 Android	V1.0(1)	30.35M	cn.peimin.testme	2020-10-28 15:40:14	下载 刷除
解决方案		testme	i Android	V1.0(1)	29.58M	cn.peimin.testme	2020-10-27 20:16:08	下载 删除
计费相关		testme	📥 Android	V1.0(1)	29.57M	cn.peimin.testme	2020-10-27 19:52:52	下载 删除
帮助		testme	🛎 Android	V1.0(1)	33.11M	cn.peimin.testme	2020-10-27 19:07:31	下载 删除

3: 在上传应用右侧栏, 单击热区选择应用程序, 或将应用程序拖动到热区上方, 上传应用。

上传应用



? 说明

- 应用程序的文件大小不大于2G。
- iOS应用程序需上传开发模式下的Debug包,即xcode for development。
- 用于检测隐私合规的应用需上传未经加固的.apk格式的Android应用包。

应用上传后,返回**应用管理**页面,查看已上传的应用。

3.上传符号表

被混淆的应用文件,为避免测试结果异常,需先使用符号表进行解析,经过解析的文件测试后方可得到真实日志,便于问题定位。

背景信息

为防止应用安装包被反编译后泄露源代码,常常使用混淆工具对应用文件进行处理,混淆工具会使用随机字符串替换文件中实际的方法和变量名,降低代码可读性,保障 文件安全性。

同时,便于开发人员排查问题,混淆工具会生成一个符号表文件,如dSYM文件(iOS)、应用Mapping文件(Android),文件记录了源代码与混淆代码的映射关系,只 有通过符号表对错误调用栈解混淆后,才能看到真实的出错堆栈。

前提条件

- 已打开移动测试控制台页面。具体操作请参见:打开控制台页面。
- 已上传应用。具体操作请参见:上传应用。

操作步骤

- 在左侧导航栏选择测试管理 > 符号表管理。
- 2. 单击上传符号表,弹出上传符号表右侧栏。
- 3. 在上传符号表右侧栏,选择待解析的应用和版本,并单击热区选择应用程序,或拖动应用程序至热区上方,上传符号表。
- 上传成功后,在**符号表管理**页面的列表末端查看上传符号表的应用文件。



? 说明

- 符号表上传成功之后,服务自动解析对应的应用,解析完成后,应用由默认的待解析状态更新为解析完成。
- 单击操作列的更新,更新指定应用的符号表。
- 单击操作列的删除,确认后删除指定应用的符号表。

4.管理测试用例

4.1. 打开用例管理页面

前提条件

- (可选)已上传测试应用。具体操作参见:上传应用。
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面。

操作步骤

- 1:在移动测试控制台页面,打开**测试管理 > 用例管理**页面。
- 2:在应用下拉列表,选择测试应用。

? 说明

应用需预先上传至移动测试控制台;否则,单击**上传应用**按钮,即刻上传。具体操作参见:<mark>上传应用</mark>。

3: 单击脚本管理/用例库/参数池/自定义步骤子页签,进行相应操作。



具体说明如下:

子页签	说明
脚本管理	用于在线录制/上传脚本,并对脚本进行管理。
用例库	用于新建用例,并对用例按照用例集/模块/用例库版本的组织层次进行管理。
参数池	用于管理录制类型脚本的相关参数。
自定义步骤	用于编写代码,自定义执行步骤。

4.2. 组织层次说明

测试用例的组织层次如下图所示:



术语	说明
用例库	用例库基于版本进行隔离管理。用例库版本建议与应用的研发版本相对应。
模块	模块用于结合应用的业务功能,对用例集做进一步组织。例如,首页/我的/订单,均为常用模块。
用例集	用例集是最小可执行的测试单元。 用例集由1个或多个用例组成,同时选择相应的参数匹配不同环境。 用例集可传入形参和实参。
用例	用例由 脚本 与所需 参数 共同构成。 用例可被不同用例集引用,作为单步骤使用。

4.3. 管理脚本

4.3.1. 新建脚本目录

前提条件

- (可选)已创建指定应用的用例库版本。具体操作参见:新建用例库版本。
- 已打开指定应用的用例管理页面。具体操作参见:打开用例管理页面。

操作步骤

- 1:在指定应用的用例管理页面,选择脚本管理页签。
- 2:从用例库版本下拉列表,选择脚本所在用例库版本,或使用默认版本。

⑦ 说明 用例库版本需预先设置。具体操作参见:新建用例库版本。



3:如首次创建目录,单击新建目录按钮;或从列表快捷菜单中,选择新建目录选项,打开新建目录对话框。 配置说明如下:

属性	说明
父级目录	 用于指定目录的存储路径,即所在父级目录。 ② 说明 ② 父级目录可选择已创建的目录或根目录。 ● 支持输入关键字,模糊匹配目录名称,搜索指定目录。
目录名称	用于设置目录的名称。支持使用中文/英文大小写/数字。

4: 单击确定按钮, 在指定父级目录下创建目录。

脚本管理	用例库	参数池	自定义步骤
0.0.1			
	ST		

4.3.2. 新建/导入脚本

前提条件

- (可选)已创建指定应用的用例库版本。具体操作参见:新建用例库版本。
- (可选)已创建脚本目录,具体操作参见:新建脚本目录。
- (可选)已在其他用例库版本创建脚本。
- 已打开指定应用的用例管理页面。具体操作参见:打开用例管理页面。

新建脚本

- 1:在**用例管理**页面,选择**脚本管理**页签。
- 2:从用例库版本下拉列表,选择模块所在的用例库版本,或使用默认版本。

? 说明

用例库版本需预先设置。具体操作参见:<mark>新建用例库版本</mark>

```
3:如右侧无数据显示,单击新建脚本;或从列表快捷菜单,选择新建脚本选项,打开新建脚本右侧栏。
```

🛎 手机淘宝 🛛 🗸 🚺 上	上传应用
脚本管理 2月库 参数池 自定	主义步骤
1.0.0 🗸 🗄 5	
3 D WLDTEST 新建脚本 6	
导入脚本	
新莲日求	
	选择脚本查看或者新建脚本
	您可尝试录制脚本或者上传脚本来新建
	4 新建脚本 导入脚本 在线录制
• 左 乾建脚士 左侧锉 诘尼脚太相关/;	信自

新建脚本	×
脚本名称:	
WLDTEST脚本	
脚本描述:	
WLDTEST脚本	h
9放目录:	
WLDTEST	
脚本类型:	
录制脚本	

配置说明如下:

选项	说明
脚本名称	用于设置脚本的名称。 【数据类型】字符串 【取值范围】2~50字符 【字符类型】中文/英文大小写/数字/下划线/短横线 【是否唯一】是
脚本描述	用于添加脚本的说明信息,便于后续查询回顾。
存放目录	用于设置保存脚本的目录。 ② 说明 脚本目录需预先设置,或保存至根目录。具体操作参见:新建脚本目录
脚本类型	用于设置脚本的类型,包括: • 录制脚本:通过在线录制功能生成的测试脚本。 • 上传脚本:用户编写的测试脚本。当脚本类型设置为上传脚本,显示上传脚本选项。
上传脚本	当脚本类型设置为上传脚本,用于选择并上传脚本文件。 ⑦ 说明 脚本编写说明参见:编写Android测试脚本/编写iOS测试脚本测试 脚本文件须打包为.zip文件。目前仅支持上传.zip类型文件。 脚本包中须包含 main.py 文件。 上传类型的脚本不可编辑。

5:单击**确定**按钮,保存脚本。

导入脚本

1:在**用例管理**页面,选择**脚本管理**页签。

2:从用例库版本下拉列表,选择模块所在用例库版本,或使用默认版本。

⑦ 说明 用例库版本需预先设置。具体操作参见:新建用例库版本

3:如右侧无数据显示,单击导入脚本;或从列表快捷菜单,选择导入脚本选项,打开导入脚本右侧栏。

🛎 手机淘宝 🛛 🗸 🗸	1> 上传应用
脚本管理 2 列库 参数池	自定义步骤
1.0.0 :	5
3 D WLDTEST 新建脚本	
导入脚本	6
新建目录	
	选择脚本查看或者新建脚本
	您可尝试录 4 本或者上传脚本来新建
	新建脚本 导入脚本 在线录制 🖂
4:在 导入脚本 右侧栏,填写脚本 [;]	相关信息。
弓 〉 脚 木	

导人脚本		×
导入目录:		
WLDTEST		
用例库版本:		
0.0.1		
选择脚本文件:		
✔ 🗁 根目录		
	Q: 允许同时导入脚本目录	

配置说明如下:

选项	说明
导入目录	用于设置保存导入脚本的目录。 ② 说明 脚本目录需预先设置,或保存至根目录。具体操作参见:新建 脚本目录
用例库版本	用于指定源脚本所在的用例库版本。
选择脚本文件	用于选择源脚本。

5: 单击确定,将指定用例库版本的源脚本,导入至当前用例库版本的指定目录。

脚本导入成功后,在左侧列表中按照组织层次显示。

				♠ 上传应用
脚本管理	用例库	参数	池	自定义步骤
0.0.1				
• 🖻 WLDTE	ST			
▼ 🖻 WLC	TEST			
Βw	/LDTEST录制	脚本		

4.3.3. 录制脚本

使用在线录制功能,通过Web页面在远程设备安装应用,录制操作步骤,并生成测试脚本。

前提条件

- (可选)已创建指定应用的用例库版本,或使用默认版本。具体操作参见:新建用例库版本。
- (可选)已创建脚本目录,或直接保存至根目录。具体操作参见:新建脚本目录。
- (可选)已新建/导入录制类型的脚本。具体操作参见:新建/导入脚本。

```
    已打开指定应用的用例管理页面。具体操作参见:打开用例管理页面。
```

安装应用

- 1. 在指定应用的**用例管理**页面,选择**脚本管理**页签。
- 2. 在**脚本管理**页签:

○ 从用例库版本下拉列表,选择模块所在的用例库版本,或使用默认版本。

⑦ 说明 用例库版本需预先设置。具体操作参见:新建用例库版本。

○ 单击在线录制,打开在线录制右侧栏。

○ 半面112333	3米前,打开1 在33米前 右侧仁。				
👅 TestM]库版本: 6.0.0	✓ 用例库版本管理 在线录	1	
	脚本管理 2		用例库	参数池	自定义步骤
脚本目	禄				
► 🗅 te	est123				
🗅 te	est11111				
🗅 si	ie				
	est123456677				
e a	est				
3 在在线录制	「右側だ・				
5. 正正规示的 i 指定品	」□	制设备			
:	利丰 洪山角洪辉 指定寻制设	·····································			
	刘衣,远中半远框,相足求制成	Ξo			
Ⅲ. 単古明	正 按钮, 打开应用安装对话框。				
(? i	说明				
北本	为 占田 的设备	于注进由			
1/1/124		., ルルマー。			
+-	* _ *1				
住家	2次前 1				
a	3.牌 华为 华为(鸿蒙) OPPO vivo 小米 小米	[1米 三星 更多			
77	転航版本 12 11 10 9 8.1.0 8.0.0 7.1	1.2 7.1.1 7.0 麗 🎓			
e e e e e e e e e e e e e e e e e e e	新聞幕 3840x1644 3200x1440 3200x1400 2700x122	.4 2.400×1080 更多			
	E chn api				
	Q. 推索品牌、初型	549个结果			
) 表尼 - XQ-BQ72(Xperia1 III) 11 3840x1644	(空闲) ^			
C	1vo - V2136A(Iqoo 8 5G) 12 1080x1920	(空闲)			
) 小米 - 2112123AC(12X) 11 1080x2400				
	vo - V2154A (Igoo Neo5S) 11 1080x2400				
2	1vo - V2157A(iqoo neo5 se) 11 1080x2400				
	▶米红米 - M2006J10C(K30 至尊纪念版) 11 1080	×2400 空闲			
	1vo - V2163A(S12 pro) 11 1080×2376				
	1vo - V2115A(T1) 11 1080x2400				
C	OPPO - PCDT10(Reno Z) 10 1080x2340	(空闲)			
	1vo - V2172A (IQOO 9 pro) 12 3200x1440				
	17:H	(SEN)	88		
4.	对话框:				
○ 単击开始	}安装 按钮,将指定应用安装到指	旨定录制设备。			
○ 单击选择	其他 按钮,打开 选择应用安装 第	对话框,从应用	列表中选择应用,将指定应	用安装到指定录制设备。	
		Ξ			
	_				
Ta	ctMa V2.0 www.mag. 2(2) 2021-09-02 14-26-17				
10	等待安装(如遏长时间无响应,清退出后重试)				
	开始安装 运择其他				
⑦ 说!	明				
本文以	单击 开始安装 按钮为例				
~~~M					
5. 应用安装完	成后,打开在线录制页面,并开	始计算在线录制	使用时长。		

⑦ 说明 在左侧的远程事	真机面板中,查看 <b>剩余体验时长</b> 和剩 <b>余资源时长</b>	• •				
<b>在线录制</b> 页面的	左侧显示已安装应用的远程真机,界面说明参见	:远程真机说明。	,			
+‡+ EVR-L29	无服务 🗋 奈 🗅 单 🔮 💦 🕅 🗑 🍋 🔲 11:33		输入	test v		保存 另存为 退出
() 08:03			清空输入	• • •		
高清			重启应用	开始录制 全部回放 刷新控件		日志 清空步骤 重置步骤
坐标			重启(清缓	脚本步骤	参数设置	控件树
			上滑			
			下滑			
	当前处于TalkBack模式。					
	长按电源键,按界面提示操作可退出。		右滑			
					暂无步骤数据	
					ngg na - na na nana per sana	
	$\vee$ $\triangleleft$ $\circ$ $\Box$					

后续如需退出在线录制,单击**退出**按钮。

## 新建脚本

- 1:在在线录制页面:
- 单击新建脚本按钮,打开新建脚本右侧栏,将录制操作存储为新脚本。
- 单击选择已有脚本按钮,打开选择脚本右侧栏,将录制操作覆盖指定脚本。

选择脚本		×
* 用例库版本:		
1.0.0		
* 选择圈本文件:		
▼ 🗌 🖻 根目录		
• 📄 🖻 WLDTEST		
🔽 🖻 WLDTEST录制脚本		
□ ▷ WLDTEST录制脚本2		
⑦ 说明		
选择的脚本需预先设置,	具体操作参见:	新建/导入脚2

## 用户指南·管理测试用例



? 说明

本文以**新建脚本**为例。

#### 2: 在**新建脚本**右侧栏,填写脚本相关信息。

新建脚本	×
* 用例库版本:	
1.0.0	
* 脚本名:	
WLDTEST在线录制脚本	
* 目录:	
WLDTEST	
* 脚本描述:	
用户指南文档开发测试用例	

## 配置说明如下:

参数	说明		
用例库版本	用于指定用例库版本号。		
脚本名	用于指定脚本的名称。 【数据类型】字符串 【取值范围】2~50字符 【字符类型】中文/大小写字母/数字/下划线/短横线		

参数	说明
目录	<ul> <li>用于指定脚本保存路径。</li> <li>⑦ 说明</li> <li>● 脚本目录需预先设置,具体操作参见:新建脚本目录。</li> <li>● 如未创建任何目录,则脚本默认保存至根目录。</li> <li>● 支持输入关键字,模糊查询已有脚本目录。</li> </ul>
脚本描述	用于添加脚本的说明信息,便于后续查询回顾。

## 3:单击确定按钮,在在线录制页面的右侧打开指定脚本的编辑器页面。 界面说明参见:脚本编辑器说明。

デーーのション・一座小小海西の

## 录制步骤

```
1: 在脚本录制页面,单击开始录制,选择按控件录制/按坐标录制,即可录制脚本。
```

```
? 说明
```

脚本录制支持**按控件录制/按坐标录制**2种方式,建议使用**按控件录制**。

2:在远程真机上使用鼠标左键单击应用,模拟屏幕点击操作,即可录制操作步骤,并在**脚本步骤**页签生成脚本步骤。

#### ? 说明

• 由于应用启动/页面加载需要一定时间,且只有在页面控件完全加载后,才能进行操作录制,所以,如涉及应用启动或页面跳转,建议增加等待步骤。

```
3: (可选)如存在交互场景,在脚本编辑工具栏选择输入,输入参数。
```

- 4: (可选)如需增加回放成功率,建议根据控件树页签的Xpath修正脚本用例。
- 5:单击**停止录制**按钮,完成操作步骤录制。
- 6:单击**全部回放**按钮*,*验证操作步骤。
- 步骤前的圆点标识:
- 绿色:回放成功。
- 红色:回放失败。
- 黄色:未执行。
- 7:单击**保存/另存为**按钮,保存脚本。
- 8:单击退出按钮,退出在线录制。



退出在线录制后,返回脚本管理页面,在左侧列表中,按照组织层次显示新建的录制脚本,并在右侧显示其内容。

🛎 TestMe	◇ 「 ● 上传应用 ● 用例』	<b>車版本:</b> 6.0.0 ∨ 用例率版本管理 ₹	已绘灵彻		
脚本管	理	用例库	参数池	自定义步骤	
脚本目录	zhangsan 🖉			<b>添加步骤</b> 录制步骤 您已规定	
<ul> <li>test123</li> <li>test11111</li> <li>cia</li> </ul>	<ul> <li>・回維的间: 2022-03-03     <li>ee     <li>・-     <li>・-     <li>・-     <li>・・     <li>・     <li>・     <li>・・     <li>・・     <li>・・     <li>・・     </li> <li>・・      <li>・・     </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li>・・      </li> <li> ・</li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></li></ul>	11:08:19 略成曲圖: 2022-03-03 11:15:18 圖本引用: 0	) 用例题目用: 0		
□ test123456677 ⊡ doc_test	a击按件: You Test Me? I Test You! 接控件点击 //android.wi cn.peimin.te You Test Me? (0.521.0.317)				
🖻 test 🖻 test111					
🖻 测试身份卡上了三… 🖻 testtest					
testtesttest zhangsan					

## 远程真机说明

远程真机用于展示在远程真机上的基本操作。



序号	按钮	控件	说明
1	系统设置工具框	终端类型	远程真机的品牌、型号。
		时长	<ul><li>远程真机的体验时长或使用资源包的时长。</li><li>远程真机的剩余体验时长和剩余资源包的时长。</li></ul>
		高清	切换设备显示屏为高清/标清显示。
		坐标	开启 <b>坐标</b> 开关键,鼠标移动至App界面上方,显示坐标值。
2	触屏操作工具框	输入	打开文本对话框,向应用的文本框输入内容。例如,用户名/密码等。
		清空输入	清空应用文本框中已输入的内容。
		重启应用	重启应用,不清理缓存。
		重启应用(清缓存)	重启应用,并清理缓存。
		上滑/下滑/左滑/右滑	滑动屏幕。
3	系统按钮	-	用于模拟Android手机的3个系统按钮。

#### 곗 说明

拖动上方+,可移动系统设置工具框。

## 脚本编辑器说明



#### 1: 状态栏

项目	说明
使用时长	显示当前在线录制的使用时长。使用时长从进入录制页面开始计时。
帮助按钮	将鼠标滑动至帮助按钮,显示当前账户使用移动测试产品的剩余体验时长和剩余资源时长。

#### 2: 保存/退出

控件	说明
保存	用于保存当前脚本。
另存为	用于将已有脚本另存为新脚本。
退出	用于退出录制页面。

#### 3: 脚本名称

控件	说明
脚本名称	用于显示当前脚本的名称。 单击下拉菜单,选择 <b>新建脚本或选择其他脚本</b> 选项,切换至其他脚本。

## 4: 工具栏

控件	说明
开始录制/停止录制	用于开始/停止录制操作步骤。

控件	说明
回放全部	用于在录制完成后,重复操作步骤。
刷新控件	用于页面切换没有及时识别到控件时刷新。
日志	用于查看客户端/服务端日志。
清空步骤	用于清空已录制的操作步骤。
重置步骤	用于重置脚本步骤为录制初始时状态。

#### 5: 页签

页签	说明
脚本步骤	用于显示录制的操作步骤。
参数设置	用于设置用户交互的输入参数。
控件树	根据控件树中的Xpath修正脚本用例,可增加回放成功率。

## 6:区间选择开关

项目	说明
开启	开启后,可按Shift键批量选择部分脚本步骤。
关闭	关闭后,按Shift键批量选择脚本步骤无效。

#### 7:步骤编辑按钮

控件	说明
单步回放	用于播放当前操作步骤。
下一行插入步骤	用于在当前操作步骤后插入步骤。
手动编辑	用于编辑当前操作步骤。
移除步骤	用于移除当前操作步骤。

## ? 说明

当鼠标滑动至脚本步骤,该步骤的序号变更为移动图标,拖动图标,可自由变更脚本步骤的顺序。

## 8: 脚本编辑工具栏

#### 脚本编辑工具栏包含6个分组:

分组	说明
常用	用于模拟终端设备的基本操作。

分组	说明
图像	<ul> <li>用于采用不同方式识别图片控件。</li> <li>⑦ 说明 <ul> <li>为提高图像识别度,建议: <ul> <li>截取屏幕图像时选择颜色对比鲜明的区域。</li> <li>OCR (Optical Character Recognition) 识别时选择字迹清晰的文字。</li> </ul> </li> <li> 若识别图像失败,建议尝试自定义脚本步骤,也可联系技术支持处理。 </li> </ul></li></ul>
自定义	用于添加自定义的代码块、步骤或脚本。
断言	用于判断控件是否存在,或者文本是否符合预期。
条件语句	用于在操作步骤中,设置条件分支。
循环语句	用于在操作步骤中,设置循环分支。

## 后续操作

```
新建用例
```

## 4.3.4. 编写脚本

## 4.3.4.1. 编写Android测试脚本

移动测试支持执行遵循Appium测试框架编写的自动化测试代码。

## 🗘 注意

移动测试平台目前可支持pytest、unittest和robotframework等三方库,如需其他支持,联系技术支持。

## 前提条件

- 了解使用Appium进行自动化测试的方法。具体说明请参见:Appium官方文档。
- 具备Python编程基础。

#### 创建Capabilities文件

使用Python语言编写Capabilities文件,指定执行测试脚本所需测试环境,文件命名为 desired_capabilities.py 。

```
其中包括 get_uri() 和 get_desired_capabilities() 2个函数。
```

函数	说明
<pre>get_desired_capabilities()</pre>	获取本次会话的参数。 参数可根据实际需求进行设置。具体说明请参见:Appium Desired Capabilities。
get_uri()	返回Appium Server的地址。

### 示例代码

#!/usr/bin/env python

def get_desired_capabilities(): desired_caps = { 'platformName': 'Android', 'platformVersion': '7.0', 'deviceName': 'V889F', 'appPackage': 'com.alibaba.mts.mtsdemoapp', 'appWaitPackage': 'com.alibaba.mts.mtsdemoapp', 'automationName': 'Appium' } return desired_caps

def get_uri():

return "http://localhost:50000/wd/hub"

#### 示例说明

• 在 get_desired_capabilities() 函数中设置了8个参数:

参数	说明
platformName	测试设备的系统类型。可选Android/iOS平台。
platformVersion	测试设备的系统版本。
deviceName	测试设备的名称。
appPackage	待测应用的包名。仅适用于Android平台。
appWaitPackage	测试时会等待。仅适用于Android平台。
арр	待测应用安装文件的存储位置。
newCommandTimeout	两条指令之间允许的最长时间间隔。 如两条指令之间的时间间隔超过预设取值,则Appium Server将终止本次会话。
automationName	本次会话所使用的自动化引擎。 • Android 4.2以下:设置为 Selendroid 。 • iOS/Android 4.2以上(含):设置为 Appium 。

• 本地启动的Appium Server,监听端口默认设置为50000,则返回值为 http://localhost:50000/wd/hub 。

#### ? 说明

desired_capabilities.py 仅用于本地执行/验证测试脚本,待测试脚本打包上传后,移动测试平台自动生成并替换该文件。

## 创建入口脚本文件

使用Python语言编写测试脚本,作为执行自动化测试的入口,文件命名为 main.py 。

示例代码及说明如下:

## 移动测试

# -*- coding: utf-8 -*from appium import webdriver # 引入刚刚创建的同目录下的desired capabilities.py import desired_capabilities # 我们使用python的unittest作为单元测试工具 from unittest import TestCase # 我们使用python的unittest作为单元测试工具 import unittest # 使用time.sleep(xx)函数进行等待 import time class MqcTest(TestCase): global automationName def setUp(self): # 获取我们设定的capabilities, 通知Appium Server创建相应的会话。 desired_caps = desired_capabilities.get_desired_capabilities() # 获取server的地址。 uri = desired_capabilities.get_uri() # 获取使用的测试框架 self.automationName = desired_caps.get('automationName')
# 创建会话,得到driver对象,driver对象封装了所有的设备操作。下面会具体讲。 self.driver = webdriver.Remote(uri, desired caps) def test_searchbox(self): # 找到包含"Tab4"字符串的控件。 if self.automationName == 'Appium': tab4 = self.driver.find_element_by_name("Tab4") else: tab4 = self.driver.find_element_by_link_text("Tab4") # 点击. tab4.click() # 等待2秒钟 time.sleep(2) # 通过控件类名找到用户名和密码输入框。 editTexts = self.driver.find elements by class name("android.widget.EditText") # 第一个框为用户名输入框,输入用户名; 第二个框为密码框,输入密码 editTexts[0].send_keys("admin") editTexts[1].send_keys("admin") # 隐藏出现的软键盘 self.driver.hide_keyboard() # 找到包含"登录"的按钮并点击 if self.automationName == 'Appium': self.driver.find_element_by_name("登录").click() else: self.driver.find_element_by_link_text("登录").click() # 等待3秒钟,登录需要与服务器通讯。 time.sleep(3) def tearDown(self): # 测试结束,退出会话。 self.driver.quit() if __name__ == '__main__': unittest.main()

## ? 说明

移动测试平台通过执行 main.py 文件的测试脚本实施功能测试。

### 启动Appium Server

在本地命令行,执行 appium 命令,启动Appium Server。例如: appium -p 50000 。

#### 参数释义

参数	说明
-p	指定Appium Server监听的端口号。

参数	说明
-a	指定Appium Server的IP地址。
-selendroid-port	在Selendroid模式下,指定adb forward的端口。
-U	当Appium Server挂载多个设备时,指定用于测试的设备的串号。
full-reset	设置测试用例执行完成后,对设备进行完全清理。

## 执行测试脚本

在本地命令行,执行 python main.py 命令,执行并验证测试脚本功能正常。

#### 提交至移动测试

1: 将测试文件打包为zip文件。

```
? 说明
```

main.py文件必须放在根目录下。

2: 在移动测试控制台,新建上传脚本,用于后续的Android功能测试。具体操作请参见:新建脚本。

## 4.3.4.2. 编写iOS测试脚本

移动测试支持执行遵循Appium测试框架编写的自动化测试代码。本文介绍如何使用Appium对iOS 10设备进行功能测试的方法。

#### ↓ 注意

移动测试平台目前可支持pytest、unittest和robotframework等三方库。如需其他支持,请联系技术支持。

#### 前提条件

- 了解使用Appium进行自动化测试的方法。具体说明请参见: Appium官方文档。
- 具备Python编程基础。

## 配置环境

本文以iOS 10为例进行说明,相应环境要求如下:

- macOS 10.11.5及以上
- Xcode 8.0及以上
- Appium 1.6.0及以上

#### 创建Capabilities文件

使用Python语言编写Capabilities文件,指定执行测试脚本所需测试环境,文件命名为 desired_capabilities.py 。

```
其中包括 get_uri() 和 get_desired_capabilities() 2个函数。
```

函数	说明
<pre>get_desired_capabilities()</pre>	获取本次会话的参数。 参数可根据实际需求进行设置。具体说明请参见:Appium Desired Capabilities。
get_uri()	返回Appium Server的地址。

#### 示例代码

## 移动测试

#!/usr/bin/env python

import sys

def get_desired_capabilities(): desired_caps = { 'platformName': 'iOS', 'platformVersion': '10.0', 'deviceName': 'iPhone 6s', 'udid': '36317c0f81086d7f4f99a9771179720b7962a2ad', 'realDeviceLogger':'/usr/local/lib/node_modules/deviceconsole/deviceconsole', 'app': '/Users/adam/iosapp.app', 'bundleId':'net.oschina.iosapp', 'newCommandTimeout': 60, 'automationName': 'Appium', 'noReset': True } return desired_caps def get_uri(): return "http://localhost:56000/wd/hub"

def flushio():

sys.stdout.flush()

#### 示例说明

1: 在 get_desired_capabilities() 函数中设置了10个参数:

参数	说明
platformName	测试设备的系统类型。
platformVersion	测试设备的系统版本。
deviceName	测试设备的名称。
udid	测试设备的ID。
realDeviceLogger	deviceconsole工具存储路径,用于获取测试设备的日志。
арр	应用安装程序的存储路径。 如不填写,则默认为bundleld运行手机上已安装应用的路径。
bundleid	应用的bundleid。
newCommandTimeout	两条指令之间允许的最长时间间隔。 如两条指令之间的时间间隔超过预设取值,则Appium Server将终止本次会话。
automationName	本次会话所使用的自动化引擎。 • Android 4.2以下: 设置为 Selendroid 。 • iOS/Android 4.2以上(含): 设置为 Appium 。
noReset	本次会话之前是否重置应用。

2:本地启动的Appium Server,监听端口默认设置为50000,则返回值为 http://localhost:50000/wd/hub 。

#### ? 说明

desired_capabilities.py 仅用于本地执行/验证测试脚本,待测试脚本打包上传后,移动测试平台自动生成并替换该文件。

#### 创建入口脚本文件

使用Python语言编写测试脚本,作为执行自动化测试的入口,文件命名为 main.py 。

示例代码及说明如下:

## 用户指南·管理测试用例

# -*- coding: utf-8 -*-

from appium import webdriver

# 引入刚刚创建的同目录下的desired_capabilities.py import desired_capabilities

## # 我们使用python的unittest作为单元测试工具

from unittest import TestCase

# 我们使用python的unittest作为单元测试工具 import unittest

#### # 使用time.sleep(xx)函数进行等待 import time

class MqcTest(TestCase):

#### def setUp(self):

#### # 获取我们设定的capabilities, 通知Appium Server创建相应的会话。 desired_caps = desired_capabilities.get_desired_capabilities()

- # 获取server的地址
- uri = desired_capabilities.get_uri()
- # 创建会话,得到driver对象,driver对象封装了所有的设备操作。
- self.driver = webdriver.Remote(uri, desired_caps)
- # 等待app完全加载 time.sleep(3)

#### # 第1个用例,如果检测到弹框,就点掉

def test case a dismiss alert(self):

#### while True:

- time.sleep(3)
- alertEle = self.driver.find_elements_by_class_name("XCUIElementTypeAlert") if alertEle: print 'find an alert' notAllowBtn = self.driver.find_element_by_xpath("//XCUIElementTypeButton[@label='不允许']") notAllowBtn.click()
- else:
- break

#### # 第2个用例,登录

def test_case_b_login(self):

#### # 取得导航栏的左侧按钮

leftBtn = self.driver.find_element_by_xpath("//XCUIElementTypeButton[@label='navigationbar sidebar']")
leftBtn.click()
time.sleep(3)

#### # 点击"点击头像登录"按钮

potraitEle = self.driver.find_element_by_xpath("//XCUIElementTypeStaticText[@label='点击头像登录']")
potraitEle.click()
time.sleep(3)

#### # 开始登录,输入账号密码

tfEle = self.driver.find_element_by_class_name("XCUIElementTypeTextField")
stfEle = self.driver.find_element_by_class_name("XCUIElementTypeSecureTextField")
tfEle.send_keys("mqctest".decode('UTF-8'))
time.sleep(1)
stfEle.send_keys("123456".decode('UTF-8'))
time.sleep(2)

#### # 点击`登录"按钮

loginBtn = self.driver.find_element_by_xpath("//XCUIElementTypeButton[@label='螢录']") loginBtn.click()

#### # **等待登录成功** time.sleep(5)

```
def tearDown(self):
```

#### # 测试结束,退出会话 self.driver.quit()

if __name__ == '__main__':
 try: unittest.main()
 except SystemExit: pass

## ? 说明

移动测试平台通过执行 main.py 文件的测试脚本实施功能测试。

#### 启动Appium Server

#### 在本地命令行,执行 appium 命令,启动Appium Server。例如: appium -p 50000

#### 参数释义

参数	说明
-p	指定Appium Server监听的端口号。
-a	指定Appium Server的IP地址。
-selendroid-port	在Selendroid模式下,指定adb forward的端口。
-U	当Appium Server挂载多个设备时,指定用于测试的设备的串号。
full-reset	设置测试用例执行完成后,对设备进行完全清理。

#### 执行测试脚本

在本地命令行,执行 python main.py -v 命令,执行并验证测试脚本功能正常。

#### 提交至移动测试

1: 将测试文件打包为zip文件。

```
⑦ 说明
main.py文件必须放在根目录下。
```

2:在移动测试控制台,新建上传脚本,用于后续的Android功能测试。具体操作请参见:新建脚本。

## 4.4. 管理参数池

## 4.4.1. 新建参数分组

#### 前提条件

- 已新建与应用版本对应的用例库版本。具体操作参见:新建用例库版本
- 已打开指定应用的用例管理页面。具体操作参见: 打开用例管理页面

#### 操作步骤

1:在指定应用的**用例管理**页面,选择参数池页签。

```
2: 在用例库版本下拉列表,选择参数池所在的用例库版本,或使用默认版本。
```

⑦ 说明
 用例库版本需预先设置。具体操作参见:新建用例库版本

#### 3: 单击**新建参数分组**按钮,打开**新建参数分组**对话框。

脚本管理 用例库	参数池 <b>2</b> 主义步骤	
新建参数分组	4 库版本: 1.0.0 ✓ 3	
<b>状认参数分组</b>	<ul><li>新建参数值</li><li>新建参数值</li><li>● 温馨提示:参数名仪支持 字母数字下划线。 编辑完成后请回车或点击 ✓ 进行保存。</li></ul>	导入参
	参数名	操
	電光数据	
<b>新建参数分组</b> 对i	舌框,填写 <b>参数分组名称</b> 。	

选项	说明
参数分组名称	用于设置参数分组的名称。该名称可唯一标识参数分组。 【数振类型】字符串 【取值范围】2~15任意字符 【字符类型】中文/英文大小写/数字/下划线

5:单击**确定**按钮,保存参数分组信息。

#### 新建参数分组在左侧列表显示。

📥 手机淘宝		令 上传应用	
脚本管理 用例	库参数池	自定义步骤	
新建参数分组	用例库版本:	1.0.0 $\lor$	
默认参数分组	新建参数	新達参数值 ● 温馨提示: 参数名仅支持 字母数字下划线。 编辑完成后请回车或点击 √ 进行保存。	Ę
参数分组WLDTEST	参数名		_

## 4.4.2. 新建/导入参数

## 前提条件

● 已新建与应用版本对应的用例库版本。具体操作参见:新建用例库版本	
● 已打开指定应用的用例管理页面。具体操作参见:打开用例管理页面	
新建参数	
1:在指定应用的 <b>用例管理</b> 页面,选择 <b>参数池</b> 页签。	
2:在 <b>用例库版本</b> 下拉列表,选择参数池所在的用例库版本,或使用默认版本。	
<ul> <li>⑦ 说明</li> <li>用例库版本需预先设置。具体操作参见:新建用例库版本</li> </ul>	
3:在左侧参数分组列表,选择参数所在分组。	
4:单击 <b>新建参数</b> 按钮,在下方列表增加一行。	
参数行默认包含 <b>参数名</b> 和1个参数值(即 <b>[0]</b> )。	
5:如需包含多个参数值,单击 <b>新建参数值</b> 按钮,添加参数值。	
6:单击 <b>参数名</b> 列的编辑按钮,修改并保存参数名称。	
⑦ 说明	
参数名仅支持英文大小写/数字/下划线。	
7: 单击参数值列的编辑按钮,输入并保存参数取值。	
手机淘宝 / 1>上传应用	
脚平管理 用例库 参数池 名氏文 一般	
新建参数分组 用例席版本: 1.0.0 / 3	
3         新建参数 新建参数值         6         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5         5 <t< td=""><td>导入参数</td></t<>	导入参数
参数分组WLDTEST [0]	操作
	A

### 导入参数

1:在指定应用的**用例管理**页面,选择参数池页签。

```
2: 在用例库版本下拉列表,选择参数池所在的用例库版本。
```

⑦ 说明 用例库版本需预先设置。具体操作参见:新建用例库版本	
3:在左侧参数分组列表,选择参数所在分组。 4:单击 <b>导入参数</b> 按钮,打开导入参数右侧栏。	
<b>参</b> 数导入 X	
当前操作参数分组:参数分组WLDTEST(1.0.0)	
約度:= sxcei很格中调 1 行为调 1 条参数信息。 A 列为参数名。B 列开始 为参数值(0),Cutt关键	
点面或物理选择 Excel 文件 仅支持 Jie Jie Xie 文件,仅该或第一些工作者数据	
<b>参</b> 款名	
5: 单击热区选择参数列表文件,或将参数列表文件拖至热区,批量导入参数。	
⑦ 说明	
参数列表文件需预先使用Excel编辑。编写规则如下:	
<ul> <li>每行为1个参数;每个参数可包含1个或多个参数值。</li> </ul>	
● A列为参数名。	
• B列为参数值1([0]); C列为参数值2([1])以此类推。	
A     B     C     D     E       1     parameter1     p1 value1     p1 value2	

## 参数列表导入成功后,在下方显示参数列表。

参数导.	λ	Х
当前操作	參数分组:參数分组WLDTEST(1.0.0)	
約定: e 参数值[	axcel表格中第 1 行为第 1 条参数信息, A 列为参数名,B 列开始为 0],以此类推	
×	请校验核对参数信息。您可重新读取文件或提交/ parameter 證證编辑	VIII
参数名	[0]	
username	WLDTEST	
password	WLDTEST	

## 6:单击**合并保存**按钮,将导入参数追加至参数分组,仅覆盖同名参数;或单击**覆盖保存**按钮,清空参数分组,保存导入参数。 合并保存后,导入参数追加至参数分组已有参数列表后。

📥 手机淘宝	∨ ♀ 上传应用		
脚本管理 用例	會 参数池 自定义步骤		
新建参数分组	用例库版本: 1.0.0	∨ ●	
默认参数分组	新建参数 新建参数值	● 温馨提示:参数名仅支持字母数字下划线。 编辑完成后请回车或点击 ✓ 进行保存。	导入参数
参数分组WLDTEST	参数名	[0]	操作
	paramKey1 👱	<u>/</u>	Ŭ
	username 🖊	WLDTEST 🔼	Û
	password 🖊	WLDTEST Z	Û

## 4.5. 管理用例库

## 4.5.1. 新建用例库版本

前提条件

已打开指定应用的用例管理页面。具体操作参见: 打开用例管理页面

### 操作步骤

- 1:在指定应用的**用例管理**页面,选择**用例库**页签。
- 2:从列表快捷菜单中选择用例库版本管理选项,打开用例库版本管理右侧栏。

🛎 手机淘宝			$\sim$	1 上传应用	
脚本管理	用	例库	2池	自定义步骤	
1.0.0			× 1	3	
毌 默认模	块	新建, 导入,	用例集 用例集		
		新建	模块 模块		
		用例	库版本管理	4	

## 3:在用例库版本管理右侧栏,单击新建版本按钮,打开新建版本右侧栏。

#### 4: 在新建版本右侧栏,填写用例库的版本信息。

用例库版本管理	新建版本	×
新建版本	版本号:	
版本号	1.0.0	
0.0.1	备注:	
	WLDTEST	
	导入旧版本数据(选填):	
	请选择	

#### 配置说明如下:

选项	说明
版本号	<ul> <li>用于设置用例库的版本号。</li> <li>⑦ 说明 <ul> <li>版本号为用例库版本的惟一标识,不可重复。</li> <li>版本号建议与应用的迭代版本保持一致,便于管理。</li> </ul> </li> </ul>
备注	用于添加用例库版本的说明信息,便于后续查询回顾。
导入旧版本数据	用于基于已有的用例库版本,创建新版本。

#### 5:单击**确定**按钮,保存用例库版本。

#### 返回**用例库版本管理**右侧栏,查看已新建的版本。

## 用例库版本管理 ×



返回**用例库**页签,在用例库版本下拉列表,查看已新建的版本。

🚢 手机淘宝			♀ 上传应用
脚本管理	用例库	参数池	自定义步骤
1.0.0		^ I	
1.0.0		~	
0.0.1			
# WLDT	EST1	_	

## 4.5.2. 新建/导入模块

模块用于结合应用的业务功能,对用例集做进一步组织。例如,首页/我的/订单,均为常用模块。

## 前提条件

## 移动测试

- (可选)已创建指定应用的用例库版本。具体操作参见:新建用例库版本
- (可选)已在其他用例库版本创建模块。
- 已打开指定应用的用例管理页面。具体操作参见:打开用例管理页面

#### 新建模块

- 1:在指定应用的**用例管理**页面,选择**用例库**页签。
- 2:从用例库版本下拉列表,选择模块所在用例库版本,或使用默认版本。
- ? 说明

用例库版本需预先设置。具体操作参见:<mark>新建用例库版本</mark>

3:从列表快捷菜单中选择**新建模块**选项,打开**新建模块**对话框。



#### 4:在新建模块对话框,填写模块名称。

5:单击**确定**按钮,保存模块。

#### 新建的模块在列表中显示。

📥 手机淘宝			♠ 上传应用
脚本管理	用例库	参数池	自定义步骤
1.0.0			
毌 默认模 毌 WLDTI	块 EST		

#### 导入模块

- 1:在指定应用的**用例管理**页面,选择**用例库**页签。
- 2:从用例库版本下拉列表,选择模块所在用例库版本,或使用默认版本。

```
⑦ 说明
用例库版本需预先设置。具体操作参见:新建用例库版本
```

3:从列表快捷菜单中选择导入模块选项,打开导入用例集模块右侧栏。

📥 手机淘宝	$\sim$	1 上传应用	
脚本管理用	列库 2 池	自定义步骤	
1.0.0	× 1	]	
3 m 默认模块	新建用例集 4 导入用例集		
	新建模块		
	导入模块 用例库版本管理	5	
4:在 <b>导入用例</b>	<b> 集模块</b> 右侧	栏,选择导 <i>)</i>	\的模块。
4:在 <b>导入用例</b> _{导入用例集模块}	<b> 集模块</b> 右侧	栏,选择导 <i>)</i>	∖的模块。 ×
4:在 <b>导入用例</b> 导入用例集模块	<b> 集模块</b> 右侧	栏,选择导 <i>)</i>	<b>\的模块。</b> ×
<ol> <li>在导入用例 导入用例集模块</li> <li>^{用例集模块}</li> <li>0.0.1</li> </ol>	<b> 集模块</b> 右侧	│ 栏,选择导 <i>)</i> 、	\的模块。 ×
<ol> <li>在导入用例 导入用例集模块</li> <li>^{用例集模块}</li> <li>0.0.1</li> <li>边界现块:</li> </ol>	<b> 集模块</b> 右侧	栏,选择导 <i>)</i>	\的模块。 ×
<ul> <li>4:在导入用例 · (0.0,1)</li> <li>         · (回)</li> <li></li></ul>	<b>)集模块</b> 右侧 ¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹¹	栏,选择导 <i>)</i>	\的模块。 ×
<ol> <li>在导入用例集模块</li> <li>导入用例集模块</li> <li>周/供成本:</li> <li>0.0.1</li> <li>遗择模块:</li> <li>一 健块名</li> <li>一 数以模块:</li> </ol>	<b>1集模块右侧</b> <u>titility</u> 2020-11-19 18	栏,选择导 <i>)</i> ~ 	\的模块。 ×

#### 配置说明如下:

选项	说明
用例库版本	用于指定源模块所在的用例库版本。
选择模块	用于选择导入的源模块。

#### 5: 单击确定按钮,将指定模块导入至当前的用例库版本。

导入的模块在列表中显示。

```
    ● 手机海宝
    ◇ 上角成用
    ■ か管理
    用例库
    ● 数池
    自定义步骤
    1.0.0
    「
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
    ・
```

#### ? 说明

如导入的模块与当前用例库版本已有模块同名,则自动改名,请注意区分。

## 4.5.3. 新建/导入用例集

### 前提条件

- (可选)已创建指定应用的用例库版本。具体操作参见:新建用例库版本
- (可选)已新建/导入模块。具体操作参见:新建/导入模块
- (可选)已在其他用例库版本创建用例集。
- 已打开指定应用的用例管理页面。具体操作参见: 打开用例管理页面

#### 新建用例集

- 1:在指定应用的**用例管理**页面,选择**用例库**页签。
- 2:从用例库版本下拉列表,选择模块所在用例库版本,或使用默认版本。

## ⑦ 说明 用例库版本需预先设置。具体操作参见:新建用例库版本

3:如右侧区域无数据显示,单击新建用例集按钮;或从列表快捷菜单中,选择新建用例集选项,打开新建用例集右侧栏。



新建用例集		×
类型:		
录制脚本用例	上传脚本用例	
用例均为录制类型, 用例集名称:	用例集中包含多个录制	却本用例的用例集
用例集描述(选填):		
		,
所属模块:		

#### 配置说明如下:

选项	说明
类型	用于指定用例集的类型,包括:         • <b>录制脚本用例</b> :该用例集包含1个或多个通过在线录制生成脚本的用例。         • <b>上传脚本用例</b> :该用例集包含且仅包含1个上传脚本的用例。
用例集名称	用于设置用例集的名称。 【数据类型】字符串 【取值范围】2~15个任意字符 【字符类型】中文/英文大小写/数字/下划线
用例集描述	用于添加用例集的说明信息,便于后续查询回顾。
所属模块	用于指定用例集所属模块。 ⑦ 说明 自定义模块需预先设置,或使用默认模块。具体操作参见:新建/导入模块

## 5:单击**确定**按钮,保存用例集。

新建的用例集在左侧列表按用例库组织层次显示;选中用例集,在右侧显示相关内容。

🛎 手机淘宝 🗸 🗸	↔ 上传应用					
脚本管理 用例库 参数池	自定义步骤					
1.0.0 $\vee$ :	WLD用例集 @ 录制类型					编辑
毌 默认模块						
▼ @ WLDTEST	新建用例 脚本导入 🞯					
UNDTEST1	序号 用例名称	用例描述	脚本	参数状态	操作	
E WEDIESTI						
			暂无数据			

## 导入用例集

1:在指定应用的**用例管理**页面,选择**用例库**页签。

2:从用例库版本下拉列表,选择模块所在用例库版本,或使用默认版本。

? 说明

用例库版本需预先设置。具体操作参见:<mark>新建用例库版本</mark>

3:如右侧区域无数据显示,单击**导入用例集**按钮;或从列表快捷菜单中,选择**导入用例集**选项,打开**导入用例集**右侧栏。
期本管理 用例库 <b>2</b> 加 自定义步骤	
1.0.0 🗸 🗄 5	
eP 默认模块 新建用例集	
・	
导入模块	
用例库版本管理	
选择用例集查看或者新建用例集	
新違用例集 导入用例集 4	

### 4: 在导入用例集右侧栏,填写用例集相关信息。

导入用	例集			×
用例库版本	š.:			
1.0.0				
源模块:				
WLDTE	ST			
导入至模块	ł:			
WLDTE	ST			
* 选择用例	集:			
	用例集名	4	¥注	
	WLD用例集	5	文档开发	

### 配置说明如下:

选项	说明
用例库版本	用于指定源用例集所在的用例库版本。
源模块	用于指定源用例集所在的模块。
导入至模块	用于设置将源用例集导入至当前用例库的目标模块。
选择用例集	用于指定源用例集。

#### 5: 单击确定按钮,将指定源用例集导入至当前用例库版本的目标模块。

导入的用例集在左侧列表按用例库组织层次显示;选中用例集,在右侧显示相关内容。

📥 手机淘宝	~						
脚本管理 用例库	参数池	自定义步骤					
0.0.1	× :	WLD用例集					编辑
毌 默认模块		用例管理参数集					
▼ # WLDTEST		新建用例 脚本导入 2					
<b>∃ WLD用例集</b>		序号 用例名称	用例描述	脚本	参数状态	操作	
				暂无数据			

## 4.5.4. 新建用例

### 前提条件

- (可选)已创建指定应用的用例库版本。具体操作参见:新建用例库版本
- 已新建/导入用例集。具体操作参见:新建/导入用例集
- 已新建/导入脚本。具体内容参见: 新建/导入脚本

- (可选)已新建参数分组。具体内容参见:新建参数分组
- (可选)已新建/导入参数。具体内容参见:新建/导入参数
- 已打开指定应用的用例管理页面。具体操作参见: 打开用例管理页面

### 打开用例集编辑页面

#### 1:在指定应用的**用例管理**页面,选择**用例库**页签。

2:从用例库版本下拉列表,选择模块所在用例库版本,或使用默认版本。

#### ? 说明

用例库版本需预先设置。具体操作参见:<mark>新建用例库版本</mark>

3:从左侧用例库组织层次列表中,选择用例集,在右侧显示指定用例集的编辑页面。

<ul> <li>▲ 手机淘宝</li> <li>✓</li> <li>✓</li> <li>// 御本管理</li> <li>用例库</li> <li>(2)</li> </ul>	<ol> <li>上传应用</li> <li>自定义步骤</li> </ol>					
1.0.0 3	WLD用例集         ②制类型           用例管理         参数集         5           新達用例         脚本导入         ④					编辑
図 WLDTEST上住脚本用例集 図 WLD用例集 ④ WLDTEST1	序号 用例名称	用例描述	<b>脚本</b>	参数状态	操作	
						<b>—</b>

#### 곗 说明

- 用例集名称右侧显示当前用例集的类型,即录制类型/上传类型。
- 录制类型的用例集可包含1个或多个通过在线录制生成脚本的用例。
- 上传类型的用例集仅可包含1个上传脚本的用例。
- 用例集的编辑页面包括用例管理和参数集2个页签。
  - 用例管理:用于在录制类型/上传类型的用例集,编排用例。
  - 参数集:用于在录制类型的用例集,设置用例相关参数。

## 新建用例

1:在指定用例集的编辑页面,选择用例管理页签。

#### 2: 单击新建用例按钮, 打开新建用例右侧栏。

🛎 手机淘宝 🗸 🗸	♀ 上传应用					
脚本管理 用例库 参数池	自定义步骤					
1.0.0 🗸 🗄	WLD用例集 @ 录制类型					编辑
毌 默认模块	用例管理数集					
	新建用例 2本导入 ②					
B WLDTEST上传脚本用例集	序号 用例名称	用例描述	脚本	参数状态	操作	
<b>図 WLD用例集</b>						
⊕ WLDTEST1			暂无数据			

3: 在新建用例右侧栏,填写用例相关信息。

新建用例	×
用例名称:	
WLDTEST用例	
用例描述:	
文档开发测试用例	li.
选择概本:	
B WLDTEST录制脚本	
前提条件(选填):	
已打开功能页面	11
期望结果(选填):	
正确记录数据	
	88

## ^{确定}^{取涨} 配置说明如下:

选项	说明
用例名称	用于设置用例的名称。如未设置,选择脚本后,默认设置为脚本的名称。
用例描述	用于设置用例的说明信息,便于后续查询回顾。
选择脚本	用于在当前用例库版本的组织层次中,选择1个已创建/导入的脚本。 ⑦ 说明 <ul> <li>● 脚本需预先创建/导入,具体内容参见:新建/导入脚本</li> <li>● 仅支持选择1个脚本。</li> </ul>
前提条件	用于说明执行当前用例的前提条件。
期望结果	用于说明执行当前用例的期望结果。

### 4:单击**确定**按钮,保存用例。

返回用例集的**用例管理**页签,在列表中查看已创建的用例。

🛎 手机淘宝 🗸 🗸	↔ 上传应用					
脚本管理 用例库 参数池	自定义步骤					
1.0.0 🗸 🗄	WLD用例集 ◎ 录制类型					编辑 下载
毌 默认模块	用例管理参数集					
▼ # WLDTEST	新建用例 脚本导入 ②					
ULD用例集 ULDTEST1	序号 用例名称	用例描述	脚本	参数状态	操作	
	1 WLDTEST用例	文档开发测试用例	WLDTEST录制脚本	无需参数	编辑用例   删除	

## 批量导入脚本

**录制类型**的用例集,可包含1个或多个通过在线录制生成脚本的用例。如需包含多个用例,可批量导入脚本,生成用例。

```
注意
脚本导入功能仅适用于录制类型的用例集。
```

```
1:在指定录制类型用例集的编辑页面,选择用例管理页签。
```

```
2: 单击导入脚本按钮, 打开脚本批量导入用例右侧栏。
```

🛎 手机淘宝		↔ 上传应用					
脚本管理  用例库	参数池	自定义步骤					
1.0.0	× :	WLD用例集   →	)				编辑
毌 默认模块		用例管理 1 集					
▼ # WLDTEST		新建用例 脚本导入					
回 WLDTEST上传题	1本用例集	序号 用例名称	用例描述	脚本	参数状态	操作	
■ WLD用例集							
				暂无数据			

3: 在脚本批量导入用例右侧栏,选择1个或多个录制类型的脚本。

脚本批量导入用例	×
选择脚本:	
<ul> <li>● 局局录</li> <li>● WLDTEST</li> <li>● WLDTEST录初脚本</li> <li>● WLDTEST录初脚本2</li> </ul>	

已选择: WLDTEST录制脚本 WLDTEST录制脚本2

4:单击**确定**按钮,将指定的1个或多个录制类型的脚本,作为用例添加至用例集。 返回用例集的**用例管理**页签,在列表中查看已批量导入的脚本的用例。

📥 手机淘宝	~	命 上传应用				
脚本管理    用例库	参数池	自定义步骤				
1.0.0	× 1	WLD用例集 ❷ 录	制类型			编辑下载
毌 默认模块		用例管理参数集				
▼ # WLDTEST		新建用例 脚本导入	0			
図 WLD用例集		序号 用例名称	用例描述	脚本	参数状态	操作
₽ WLDTEST1		1 WLDTEST录制	脚本 适用于文档开发场景	WLDTEST录制脚本	无需参数	编辑用例   删除
		2 WLDTEST录制	脚本2 文档开发使用	WLDTEST录制脚本2	无需参数	编辑用例 删除

## 添加用例集参数

囗 注意

添加用例集参数功能仅适用于录制类型的用例集。

1:在指定**录制类型**用例集的编辑页面,选择**参数集**页签。

2: 在参数集页签,选择已创建的参数分组,或使用默认分组。

? 说明

- 参数分组需预先设置。具体操作参见∶<mark>新建参数分组</mark>
- 如指定的参数分组已创建/导入参数,则添加为用例集的公共参数。

3: 单击**新建参数**按钮,添加用例集参数; 单击**新建参数值**按钮,添加用例集参数的取值。具体操作同:新建参数

🛎 手机淘宝		① 上传应用		
脚本管理 用例	库 参数池	自定义步骤		
1.0.0	× :	WLD <b>用例集                                    </b>		编辑
毌 默认模块 ▼ 毌 WLDTEST 図 WLDTEST上 図 WLD用例集	传脚本用例集	参数分组:         参数分组WLDTE           用例集参数:         ●           新建参数         新建参数	EST         2           2         3           3         温馨提示:参数名仅支持 字母数字下划线。编辑完成后请回车或点击	
₽ WLDTEST1		参数名 paramKey1 ∠		操作
		公共参数:		
		参数名 paramKey1	[0]	dt
		username password	WLDTEST WLDTEST	

# 5.实施Android测试

# 5.1. 兼容测试

## 5.1.1. 提交测试

## 前提条件

- 已购买基础测试资源包,或使用量未超过免费阈值。具体说明及操作参见:产品定价。
- (可选)已上传待测试的Android应用。具体操作参见:上传应用。
- 如应用需登录后启动,需预先设置登录的用例集和参数分组。具体操作参见:新建/导入用例集/新建参数分组。
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面。

## 操作步骤

1:在移动测试控制台的左侧导航栏,选择**测试产品 > 兼容测试**页签。

#### 2:在应用下拉列表选择待测试的Android应用。

#### ? 说明

```
待测试的Android应用需预先上传;如未上传,单击上传应用按钮,即刻上传。具体操作参见:<mark>上传应用</mark>。
```

## 3: 单击提交测试,打开运行兼容测试页面。

移动测试	
测试产品 1	▲ · · · · · · · · · · · · · · · · · · ·
兼容測试 2	<b>3.</b> 宽 测试报告 4
功能测试性能测试	应用販素: VA 5 1028/1020/2020_11_11 1 4-5-(10) √
远程真机	
测试管理 🖌 🖌	
解决方案 🖌 🖌	
计费相关	
帮助	新王卿过招告
	请点击提交测试马上进行提测
4: 在运行兼容测试员	页面,按步骤提示,对测试进行配置。具体内容参见: <mark>配置说明</mark> 。
5:配置完成后,将鼠	标移动至 <b>提交测试</b> ,显示当前测试预计消耗资源包。
预计消耗资源包 294 台次 >> 握交测试成功即消耗资源包, => 上一步 提交测试	>>购货资源包 无法退回或取消 取3消
⑦ 说明 预计消耗资源包如	超过当前账户的免费阈值,或超过已购资源包限额,单击 <b>购买资源包</b> 链接,购买 <b>基础测试</b> 资源包。
6:单击 <b>提交测试</b> ,提	<b>之</b> 须试。
选择应用及版本	测试配置 3 完成
	✓ 提交测试成功
	平台当前还 <b>有 32</b> 个任务等待执行,请耐心等待,执行完毕后可查看完整 测试报告。
	返回兼容性测试
测试提交后,排队等待	持执行;测试执行完毕后,可查看完整测试报告。
配置说明	
选择应用及版本	

## 用户指南·实施Android测试

AA6-White   ACCS      A       A       A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A      A	() RARE () SE					
T-# Ris						
选项	说明					
应用包	默认选中已上传/选择的应用,可修改。					
应用版本	默认选中已上传/选择的应用的版本。如应用存在多个版本,则选择用于兼容性测试的版本。 ⑦ 说明 若待测试的应用为混淆版本,请先上传符号表,应用解析成功后再提交测试。上传符号表的具体操作请参见上传符号表。	默认选中已上传/选择的应用的版本。如应用存在多个版本,则选择用于兼容性测试的版本。 ⑦ 说明 若待测试的应用为混淆版本,请先上传符号表,应用解析成功后再提交测试。上传符号表的具体操作请参见上传符号表。				
测试备注	输入测试相关信息,便于后续查询回顾。					
<ul> <li>測试配置</li> <li>送择应用及版本</li> <li>基础信息</li> <li>正式版 体验版</li> </ul>	2 测试配置	3 完成				
<ul> <li>服务项</li> <li>自定义测试机型</li> <li>错误日志、自动截图</li> <li>Ripper智能探索</li> <li>日、洗茶和取</li> <li>P.洗径 0.数</li> </ul>	<ul> <li>检测项</li> <li>计费项</li> <li>○ 应用安装、启动、卸载</li> <li>◇ 按量付费,不限时间及次数</li> <li>◇ 启动时间、CPU占用率、内存占用率</li> <li>◆ 用户登录、流量</li> </ul>					
<ul> <li>測试配置</li> <li>測试时长:         <ol> <li>10</li> <li>登录方式:                 <li>元需登录 用例集登录</li> <li>邮件通知地址:                 </li> <li>邮件通知地址:                 </li> <li>邮件通知地址:                 </li> <li>単件通知地址:                 </li> </li></ol></li></ul>						
<b>基础信息</b> 分组包括:						
选项	说明					
正式版/体验版	<ul> <li>正式版:已购买基础测试资源包,付费使用的版本。         <ol> <li>证明 使用正式版运行兼容性测试,需购买基础测试资源包。</li> </ol> </li> <li>体验版:使用量在免费阈值内,体验使用的版本。         <ol> <li>体验版:使用量在免费阈值内,体验使用的版本。</li> <li>说明 在体验版的计费项列,可查看当前账号的免费额度;测试超出的使用量,需按量付费。</li> </ol> </li> </ul>					

选项	说明
选择机型	1: 单击选择机型,打开选择提测机型右侧栏,手动指定或通过预选项批量指定预选机型。 2####N型 ***********************************

## **测试配置**分组包括:

选项	说明
测试时间	如使用 <b>正式版</b> ,则设置测试的时间长度。 ⑦ 说明 如使用体验版,则测试时间默认为10分钟,不可设置。
登录方式	<ul> <li></li></ul>
用例库版本	当 <b>登录方式</b> 设置为用例集登录,指定登录用例所在的用例版本库。 ⑦ 说明 用例库版本需预先设置,或使用默认版本。具体操作参见:新建用例库版本。
测试用例集	当登录方式设置为用例集登录,指定登录用例所在的用例集。 ⑦ 说明 登录用例集需预先设置;如未预先设置,单击用例管理按钮,打开测试管理 > 脚本管理页面,即刻进行设置。具体操作参见:新建/导入用例 集。
参数分组	当 <b>登录方式</b> 设置为 <b>用例集登录</b> ,指定登录用例集相关参数分组。 ⑦ 说明 登录用例集相关参数分组需预先设置,具体操作参见:新建参数分组。
账号互踢	当 <b>登录方式</b> 设置为 <b>用例集登录</b> ,设置应用账号是否仅限在唯一终端登录。

 选项
 说明

 邮件通知地址
 设置电子邮箱地址,用于接收测试完成通知。

后续操作

查看概览信息

查看测试报告

## 5.1.2. 查看概览信息

## 前提条件

- 已提交兼容测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

#### 操作步骤

- 1:在移动测试控制台的左侧导航栏,选择**测试产品 > 兼容测试**页签。
- 2:在应用下拉列表选择已提交测试的Android应用,默认打开概览页面。

移动研发平台EMAS / 移动	加测试 / 兼容测试						
移动测试							
<ul> <li>測试产品</li> <li>兼容測试</li> <li>力能測试</li> <li>性能測试</li> <li>近程真机</li> </ul>	<ul> <li>▲ 手机淘宝</li> <li>④ 提交測试</li> <li>● 上传</li> <li>概 览</li> <li>④ 报告</li> <li>应用版本:</li> <li>V9.15.0(304)(2020-11-19 18:46:43) √</li> <li>⑤</li> </ul>	应用			6	7	
测试管理 ^	Android兼容性测试 (已完成)			提测于:	2020-11-25 11:30:56 🛛 🗸	■ C	Ł
用例管理	测试结果 错误分布	CRASH 设	备数 次数	ANR		设备数	次数
任务管理				0,14256,c	om.taobao.taobao,9495015	1	2
应用管理 解决方案 <b>^</b>				0,28710,c	om.taobao.taobao,1207456	1	1
专家服务 (Hot)		斯无数据		0,9394,co	m.taobao.taobao,12074564,I	1	1
专有云服务	失败: 4						
计费相关	成功: 6						
帮助							
3:在概览页面:							
● 从 <b>应用版本</b> 下拉药	列表中选择应用的版本;						
● 从 <b>提测于</b> 下拉列表	<b>長中,选择提测时间,默认为最近一次提测;</b>						
查看指定应用/版本/	/提测时间的测试概览信息。具体内容参见: <mark>概览页</mark>	面说明					
4: 单击 <b>查看报告</b> 按	钮,打开指定测试的详情页面。具体内容参见:测试	试报告说明					
5: 如当前测试状态:	为 <b>运行中</b> ,单击 <b>刷新</b> 按钮,手动刷新 <b>概览</b> 页面,查	看最新信息。					

6: 单击**下载报告**按钮, 下载指定测试的测试报告。

## ? 说明

如当前测试状态为**运行中**,则**下载报告**按钮非活性。

## 概览页面说明



## Android兼容测试概览页面包括:

信息	说明
测试状态	显示当前测试的状态,包括 <b>运行中</b> 和 <b>已完成</b> 。
测试结果	显示当前测试中运行中/失败/未执行/取消的设备数,及所占比例。
错误分布	显示当前测试中发生的错误类型、影响设备数,及各类错误所占比例。
CRASH	显示当前测试中崩溃异常的统计数据,包括影响设备数和次数。
ANR	显示当前测试中无响应异常的统计数据,包括影响设备数和次数。

## 5.1.3. 查看测试报告

#### 前提条件

- 已完成兼容测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面

## 查看测试报告

- 1:在移动测试控制台的左侧导航栏,选择测试产品 > 兼容测试页签。
- 2:在应用下拉列表选择已测试的Android应用,选择测试报告页签,查看指定应用执行的兼容测试列表。

移动研发平台EMAS /	移动	则试 / 兼容测试						
移动测试		3						
测试产品 🚺 🤸		📥 手机淘宝	∨ ▷ 提交測试	↔ 上传应用				
兼容測试         2           功能測试		概 览 测试报告	4					
性能测试 (New)		C RIM 5						
远程真机		编号	应用版本	提測时间	測试结果 🛛	状态	▼ 备注	操作
测试管理 解决方案		718476	9.15.0	2020-11-25 11:30:56	10 (6 / 4 / 0)	● 已完成	WLDTEST: Android兼	容 🖉 🧰 👌
计费相关								〈 上一页 】 下一页 〉
帮助								

3:如指定应用有运行中的测试,单击**刷新**按钮,手动刷新**测试报告**页面,查看最新信息。

4:在测试报告列表的操作列:

- 单击查看按钮,查看指定测试的测试报告页面。具体内容参见:测试报告说明
- 单击分享按钮, 生成指定测试的测试报告的分享链接。通过分享链接, 任意用户均可使用浏览器查看该测试报告。
- 如测试正在运行中,单击**取消**按钮,中止测试。

## 5.1.4. 测试报告说明

## 概述

(已完成) 下载 分享 2

移动研发平台EMAS / 移动测试 / 任务管理 / 测试报告详情

← 手机淘宝(V9.15.0(304)) 
●

概况 设备详情 错误日志 性能报告 全部截图 1

#### 1:选择各页签查看相关内容。

页签	说明
概况	查看当前测试的执行结果、失败类型、失败机型分布、错误日志、失败机型概况。
设备详情	查看当前测试使用的机型,以及测试项目的执行进度和执行结果。
错误日志	查看当前测试的报错信息,包括 <b>崩溃、卡死</b> 和和其他 <b>可忽略异常</b> 。
性能报告	查看当前应用对设备资源占用情况,包括 <b>冷启动耗时、FPS、CPU/内存平均占用率</b> 等。
全部截图	查看当前测试的关键截图。

#### 2: 单击右上角各按钮,进行相关操作。

按钮	说明
下载	下裁当前测试报告。
分享	生成当前测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。

#### 3: 鼠标移动至帮助按钮,查看当前测试基本信息。

D: 17136 測试类型: Android兼容性測试 提測人: m************* 提測时间: 2020-11-13 15:59:57

## 概况

#### 在概况页签,查看当前测试的执行结果、失败类型、失败机型分布、错误日志、失败机型概况。

执行结果

### 显示**通过/未通过/异常未执行**的机型数及其占比。

执行结果			
	类型	占比	机型数量
机型总数	通过	60%	6
10	未通过	40%	4
	异常/未执行	0%	0

#### 失败类型

显示安装失败/启动失败/运行失败的机型数及其占比。

失敗类型			
	类型	占比	机型数量
机型总数	安装失败	25%	1
<ul> <li>运行失数: 76%</li> </ul>	启动失败	0%	0
	运行失敗	75%	3

### 失败机型分布

按照指定维度,指定状态,显示数量及占比。

失敗机型分布	1	品牌	~	全部状态	$\sim$
509/		品牌	~	2	
30%		CPU			
1	1	GPU			
25%	25%	系统版本			
		内存			
		分辨率			
1870 Mataraia	16.25	屏幕			
	+ w		_		

选项	说明
维度	按照指定维度,显示各状态的设备数量及占比。 维度包括:品牌/CPU/GPU/ <b>系统版本/内存/分辨率/屏幕</b> 。
状态	显示指定维度,指定状态的数量及占比。 状态包括:全部状态/安装失数/启动失数/运行失数。

## 错误日志

#### 显示关键错误的列表。

#### 错误日志

错误类型	出現次数	关键错误	操作
卡死	2	0,14256,com.taobao.taobao.9	详情
卡死	1	0,9394,com.taobao.taobao,12	详情
卡死	1	0,28710,com.taobao.taobao,1	详情
可意略异常	21	java.lang.ClassNotFoundExce	详情

## □ 查看全部

单击**查看全部**按钮,或者单击**详情**按钮,打开**错误日志**页签,查看全部错误及其日志详情。具体内容请参见:错误日志。

#### 失败机型概况

显示各失败机型的测试结果。

失败机型概况								
<u>и</u> а (1	安装	启动	$\overline{\mathbf{T}}$	自动探索	崩溃	卡死	可忽略异常	操作
华为 KIW-CL00 6.0.1	0	0		8	8	1	0	详情
华硕 ASUS_Z00ADB 5.0	•	8		• 2	2	1	• 3	详情
360 1501_M02 (F4) 5.1	0	0		8	8	2	0	详情
OPPO PCAM10(A9) 9	0	0		8	5	6	0	详情
OPPO PCLM50 (K7) 10	•	0		8	2	2	1	详情
vivo X6D 5.1	•	0		0	7	1	0	详情
三星 SM-A9100 (A9) 6.0.1	0	0		8	4	0	0	详情
中兴 BA910 (BladeV9) 5.1	•	0		8	5	1	0	详情
乐视 Le X625 (乐2Pro) 6.0	•	0		0	5	1	0	详情
华为 CAM-TL00H (荣耀畅玩5A) 6.0	•	0		8	5	1	1	详情
							<	1 2 >

1:在列表的表头,单击**安装/启动/自动探索**筛选按钮,对列表内容进行筛选。

2: 在崩溃/卡死列,显示设备的异常个数,单击数字链接,查看异常的详细信息。

3: 在操作列, 单击**详情**按钮, 打开**设备详情**页面。具体内容请参见: 设备详情。

### 设备详情

在**设备详情**页签:

1:在左侧**设备**列表,单击**筛选**下拉按钮,按照**分类方式**或状态筛选设备。

筛选项	说明
分类方式	将测试设备按照指定分类方式进行分组,包括 <b>按测试结果、按系统版本、按品牌</b> 3个选项。
状态	将测试设备按照指定状态进行筛选,包括 <b>全部、未通过、设备错误、异常/未执行、超时、通过、等待中、运行中</b> 。 当 <b>状态</b> 设置为 <b>未通过</b> 选项,可进一步按照 <b>失败环节</b> 进行筛选,包括 <b>安装、启动、运行、卸载</b> 4个环节。

2: 在左侧设备列表,单击设备的名称,在右侧打开指定设备的详细信息。

10.47	华为 - BAC	-AL00 (nova2 Plus) - 8.4	0.0 (未通过)			の 远程真相
设备 重跑		✓ 启动	⊗ Ripper智能探索	✓ 卸载		
◇ 未通过	分类方式:					
• Motorola - XT1581 (Moto	按测试结果(默认)					
3年为 - MT7-TL00 (Mate 7	状态:					□ 视频 设备日志
● 华为 - BAC-ALOO (nova2	全部	The second se	5 ◆ 2 ○ G ○ 未完 日 100% (第3 中午11.41)	无服务 🕈 🖸 🗖 🖬 🔹 🗼 木 宅区	110% ● 中午11:41	オ 10,0100% I=0 中年11,44 无服务 ♦ 🙆 D 🗃 💭 🕴 冬回 100% I=0 中年11,59
● 华硕 - P01V - 5.0.2		19	C #74 ···	《 外卖	… ← 吐槽吧	
∨ 通过	6			● 立即登录		<b>た</b> (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
<ul> <li>Motorola - XT1570 (Moto)</li> </ul>	X Style)			常用功能	• •	武師学 の 学学 (三)     天価中心 和解法 低加金市 阿里伯素 分表
• vivo - V1813T(Y97) - 8.1.	0	N15 00000 PHENO 770		自由48 正接关注 客服	1012	天坂田市 大山北 海空直接 11110
• VOTO - GT7 - 4.4.4	VQ.	💐 🖱 🗖		a 赚吃货豆	可及大红包 >	
● 中兴 - ZTE A2017 (天机7)	- 6.0.1 起版	手了! 眾刻第 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		更多	\$	限时施抢大期接着 来划算圆窗口 天王四
● 华为 - ELS-AN00(P40 pro)	- 10			向为会作	纳手招募	
● 小米红米 - Redmi Note 7 - :	9					
	1000 C			n E	•	
		in ha nus nunz			<b>.</b>	87 74 R072
<b>示状</b> 态						
指定设备各兼容性测	试项(安装/启动/	智能探索/卸载)的按	l行状态。			
勺 - BAC-ALOO (no						ゆ 沅程
	ova2 Plus) - 8.0.0	(未通过)				
<ul> <li>安装</li> <li>3定设备的测试未通过</li> <li>1</li>     &lt;</ul>	wa2 Plus) - 8.0.0 ● 启动 t,单击远程真机按	● (未通过) ● Ripper智能探索 钮,申请使用相同机型	● 卸载 型设备,进行测试结果	t 出验证。		
<ul> <li>安装</li> <li>定设备的测试未通过</li> <li>探索</li> <li>测试过程的关键截图</li> <li>冒能探索</li> <li>性能分析</li> </ul>	yya2 Plus) - 8.0.0 ● 启动 t,单击远程真机按 1/视频,和异常汇总	<ul> <li>◆ 未通过</li> <li>◆ Ripper智能探索</li> <li>田,申请使用相同机型</li> <li>。</li> </ul>	● 卸载 22设备,进行测试结果	^我		
安装 定设备的测试未通过 指定设备的测试未通过 指案案 示测试过程的关键截图 智能探索 性能分析 查看截图	yya2 Plus) - 8.0.0 ● 启动 1, 単击远程真机按 ⁴ 3/视频,和异常汇总	<ul> <li>★通过     <li>■ ◎ Ripper智能探索     <li>田,申请使用相同机型     </li> </li></li></ul>	● 卸载 型设备,进行测试结果	t 9金证。		设备日志
② 安装 定设备的测试未通过 能深素 能测试过程的关键截图 留能探索 性能分析 管看截图 20 ● 300	yva2 Plus) - 8.0.0 ● 启动 註, 单击远程真机按: 日/视频,和异常汇总 日/视频,和异常汇总	<ul> <li>(未通过)     <li>● Ripper智能探索     <li>钮,申请使用相同机型     </li> <li>●     </li> </li></li></ul>	● 卸载 型设备,进行测试结果	£ 9金证。	€ 309∞40	设备日志
<ul> <li>● 安装</li> <li>第定设备的测试未通过</li> <li>#探索</li> <li>※测试过程的关键截图</li> <li>#能探索 性能分析</li> <li>查看截图</li> <li>●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●</li></ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li>● 启动</li> <li>土, 单击远程真机按结</li> <li>引/视频,和异常汇总</li> <li>③/视频,和异常汇总</li> </ul>	<ul> <li>(未通过)     <li>Ripper智能探索     <li>田,申请使用相同机型     </li> <li>★0◆★400300     </li> </li></li></ul>	<ul> <li>卸载</li> <li>型设备,进行测试结果</li> <li></li> <li><td>£ 验证。 ***********************************</td><td>• ±05≈≠0</td><td>设备日志 ()))) ()) × 1100</td></li></ul>	£ 验证。 ***********************************	• ±05≈≠0	设备日志 ()))) ()) × 1100
<ul> <li>● 安装</li> <li>新定设备的测试未通过</li> <li>新探索</li> <li>新测试过程的关键截图</li> <li>雪能探索 性能分析</li> <li>查看截图</li> <li>●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●</li></ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li>② 启动</li> <li>注, 单击远程真机按结</li> <li>引/视频,和异常汇总</li> <li>③ / 视频,和异常汇总</li> </ul>	<ul> <li>(未通过)     <li>◎ Ripper智能探索     <li>钮,申请使用相同机型     </li> <li>•     </li> </li></li></ul>	● 卸载 型设备,进行测试结果 < 就應買 #m	ℓ 19金证。 ************************************	• ±09≋±ú	设备日志 ())) ()) ()) ()) ()) ()) ()) ()) ()) (
<ul> <li>◆ 安装</li> <li>定设备的测试未通过</li> <li>探索</li> <li>测试过程的关键截图</li> <li>部底探索 性能分析</li> <li>皆看截图</li> <li>●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●</li></ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li>② 启动</li> <li>其, 单击远程真机按</li> <li>引/视频,和异常汇总</li> <li>③ / 视频,和异常汇总</li> <li>③ / 视频,和异常汇总</li> </ul>	<ul> <li>(未通过)</li> <li>◎ Ripper智能探索</li> <li>钮,申请使用相同机型</li> <li>。</li> <li>*●♥ * ★ 0 0000</li> </ul>	<ul> <li>卸载</li> <li>2设备,进行测试结果</li> <li>( ¹ · ·································</li></ul>	ℓ 1验证。 ************************************	• to9≥∔u	设备日志 ② ● × 1148 亲, 欢迎登录
<ul> <li>● 安装</li> <li>二定设备的测试未通过</li> <li>深索</li> <li>测试过程的关键截图</li> <li>當能探索 性能分析</li> <li>算看截图</li> <li>●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●</li></ul>	<ul> <li>xa2 Plus) - 8.0.0</li> <li>② 启动</li> <li>其 单击远程真机按</li> <li>4/视频,和异常汇总</li> <li>3/视频,和异常汇总</li> <li>※</li> <li>※</li></ul>	<ul> <li>(未通过)</li> <li>○ Ripper智能探索</li> <li>田,申请使用相同机型</li> <li>○</li> <li>1○♥ = + 0 0000</li> <li>         型登录         1 200220     </li> </ul>	<ul> <li>卸载</li> <li>22设备,进行测试结果</li> <li>22设备,进行测试结果</li> <li>( _{東鉱具 Ma}</li> </ul>	ℓ 1號1正。 ************************************	• 10924a	设备日志 ② ● × 51x89 亲,欢迎登录 梁策策略版中2 立即注册
<ul> <li>◆ 安装</li> <li>定设备的测试未通过</li> <li>探索</li> <li>测试过程的关键截图</li> <li>常能探索 性能分析</li> <li>管看截图</li> <li>●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●</li></ul>	<ul> <li>◇ 启动</li> <li>② 启动</li> <li>2, 单击远程真机按:</li> <li>3/视频,和异常汇总</li> <li>③ ( ① ( ○ ( ○ ( ○ ( ○ ( ○ ( ○ ( ○ ( ○ ( ○</li></ul>	<ul> <li>(未通过)</li> <li>○ Ripper智能探索</li> <li>田,申请使用相同机型</li> <li>○</li> <li>*○♥ * + 0 3350</li> <li>#登录</li> <li>* 2093##</li> <li>#★###################################</li></ul>	<ul> <li>卸载</li> <li>型设备,进行测试结果</li> <li>《 報道具 #ma</li> </ul>	및 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	• 10 <b>7</b> 840	设备日志
<ul> <li>◆ 安装</li> <li>定设备的测试未通过</li> <li>深索</li> <li>测试过程的关键截图</li> <li>常能探索 性能分析</li> <li>管看截图</li> <li>● ● ● ● ● ● ● ● ●</li> <li>手机泡室</li> </ul>	<ul> <li>◇ 启动</li> <li>② 启动</li> <li>注, 单击远程真机按结</li> <li>3/视频,和异常汇总</li> <li>③ ( ① ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●</li></ul>	<ul> <li>(未通过)</li> <li>(未通过)</li> <li>(本通过)</li> <li>(本通)</li> <li>(-(-(-(-(-(-(-(-(-(-(-(-(-(-(-(-(-(-(-</li></ul>	● 卸载 型设备,进行测试结果 《 就該頁 ///////////////////////////////////	ℓ 1空 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	• • • • • • • • •	辺留日志
<ul> <li>◆ 安装</li> <li>定设备的测试未通过</li> <li>探索</li> <li>测试过程的关键截图</li> <li>■●●●●</li> <li>■●●●●</li> <li>■●●●●</li> <li>■●●●●</li> <li>■●●●●</li> <li>■●●●●</li> <li>■●●●●</li> <li>■●●●</li> <li>■●●</li> <li>■●</li>     &lt;</ul>	<ul> <li>&gt;&gt; 2 Plus) - 8.0.0</li> <li>○ 启动</li> <li>2 月 山田和</li> <li>2 月 山田和</li></ul>	<ul> <li>(未通过)     <li>● Ripper智能探索     <li>田,申请使用相同机型     </li> <li>● 400000     </li> </li></li></ul>	● 卸载 型设备,进行测试结果  < 就應買 約	ℓ 4验证。 ******	• ±09=+0	设备日志 ③ * 11x0 ● × * * * * * * * * * * * * * * * * * *
<ul> <li>● 安装</li> <li>定设备的测试未通过</li> <li>探索</li> <li>测试过程的关键截图</li> <li>留能探索 性能分析</li> <li>管看截图</li> <li>● ● ● ● ● ●</li> <li>● ● ● ● ●</li> <li>● ● ● ● ● ●</li> <li>● ● ● ● ● ●</li> <li>● ● ● ● ●</li> <li>● ● ● ● ● ●</li> <li>● ● ● ● ● ●</li> <li>● ● ● ● ●</li> <li>● ● ● ● ● ●</li> <li>● ● ● ●</li> <li>● ● ●</li> <li>● ● ● ●</li> <li>● ● ● ●</li> <li>● ● ● ●</li> <li>● ● ●</li> <li>● ● ● ●</li> <li>● ● ● ●</li> <li>● ● ●</li> <li>● ● ● ●</li> <li>● ●</li> <li>● ● ●</li> <li>● ●</li> <li></li></ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li> <ul> <li></li></ul></li></ul>	<ul> <li>◆ 未通过</li> <li>◆ Ripper智能探索</li> <li>田,申请使用相同机型</li> <li>◆</li> <li>◆ + 0 00000</li> <li>●</li> <li>●&lt;</li></ul>	<ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li>&lt; 就應具 ##</li> </ul>	ℓ 归验证。 * 91000 ****	• *0 <b>?</b> =+0	設备日志
<ul> <li>② 安装</li> <li>         第定设备的测试未通过 探察 深测试过程的关键截图 留能探索 性能分析 查看截图      </li> <li>             季4</li> <li>             季4</li> <li>             季4</li> </ul> <li>             £9599</li> £959 <li>             £9599</li> <li>             £9599</li> <li>             £9599</li> <li>             £9599</li> <li>             £959</li> <ul> <li>             £959</li>             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959             £959&lt;</ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li> <ul> <li></li></ul></li></ul>	<ul> <li>(未通过)     <li>○ Ripper智能探索     <li>田,申请使用相同机型     </li> <li>■ 0 00001     </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001     </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      </li> <li># 0 0 0001      # 0 0 0001      # 0 0 0001      # 0 0 0001      # 0 0 0001     # 0 00001      # 0 0 0001     # 0 0 0001      # 0 0 0001     # 0 00001     # 0 00001      # 0 00001      # 0 00001     # 0 00001     # 0 00001      # 0 00001     # 0 00001     # 0 00001     # 0 00001      # 0 00001     # 0</li></li></li></ul>	<ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li>&lt; 較面页 160</li> </ul>	ℓ 19金证。 ************************************	★ 10 9 ° + 0	設备日志
<ul> <li>安装</li> <li>金支装</li> <li>金定设备的测试未通过</li> <li>探索</li> <li>二、一、一、一、一、一、一、一、一、一、一、一、一、一、一、一、一、一、一、一</li></ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li></li></ul>	<ul> <li>(未通过)</li> <li>◎ Ripper智能探索</li> <li>钮,申请使用相同机型</li> <li></li> <li><!--</td--><td><ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li>&lt; 範載員 ###</li> </ul></td><td>ℓ 1號证。 **** ●</td><td>・ 109240 他に2000時2日用発色目前後日本日本年期の 日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本</td><td>送番日志 1998日 一 一 2 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 199</td></li></ul>	<ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li>&lt; 範載員 ###</li> </ul>	ℓ 1號证。 **** ●	・ 109240 他に2000時2日用発色目前後日本日本年期の 日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本日本	送番日志 1998日 一 一 2 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 二 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 1998日 199
<ul> <li>安装</li> <li>金 安装</li> <li>金 安装</li> <li>金 金装</li> <li></li></ul>	<ul> <li>wa2 Plus) - 8.0.0</li> <li> <ul> <li></li></ul></li></ul>	<ul> <li>(未通过)</li> <li>○ Ripper智能探索</li> <li>田,申请使用相同机型</li> <li>○</li> <li>10♥2+00000</li> <li>200120</li> <li>200</li></ul>	<ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li></li> <li>&lt;</li></ul>	الله تلت . الله . الم . ا . الم . الم . الم .	• <b>309≤40</b> • <b>109≤40</b> • • • • • • • • • • • • • • • • • • •	送备日志 ③ ● ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
<ul> <li>② 安装</li> <li>第定设备的测试未通过 能探索</li> <li>※測试过程的关键截图</li> <li>審能探索 性能分析</li> <li>查看截图</li> <li>●●●●●●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●●</li> <li>●●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●●</li> <li>●●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●●</li> <li>●●●●●●●●●●●</li> <li>●●●●●●●●</li> <li>●●●●●●●●</li> <li>●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●●●</li> <li>●●●●●●●</li> <li>●●●●●●●</li> <li>●●●●●●●</li> <li>●●●●●●</li> <li>●●●●●</li> <li>●●●●●</li> <li>●●●●●</li> <li>●●●●●</li> <li>●●●●●</li> <li>●●●●</li> <li>●●●</li> <li>●●</li> <li>●●<td><ul> <li>ション 2 Plus) - 8.0.0</li> <li> <ul> <li></li></ul></li></ul></td><td><ul> <li>(未通过)</li> <li>(未通过)</li> <li>Ripper智能探索</li> <li>田,申请使用相同机型</li> <li></li> <li>10マミキロ0000</li> <li>2002录</li> <li>2002年</li> <li>2014</li> <li>2014</li> <li>2014</li></ul></td><td><ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li></li> <li>&lt;</li></ul></td><td>د التكونية: •••• •••• •••• •••• •••• •••• •••• •</td><td></td><td>送名日志 1998日 一 一 一 一 二 二 二 二 二 二 二 二 二 二 二 二 二</td></li></ul>	<ul> <li>ション 2 Plus) - 8.0.0</li> <li> <ul> <li></li></ul></li></ul>	<ul> <li>(未通过)</li> <li>(未通过)</li> <li>Ripper智能探索</li> <li>田,申请使用相同机型</li> <li></li> <li>10マミキロ0000</li> <li>2002录</li> <li>2002年</li> <li>2014</li> <li>2014</li> <li>2014</li></ul>	<ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li></li> <li>&lt;</li></ul>	د التكونية: •••• •••• •••• •••• •••• •••• •••• •		送名日志 1998日 一 一 一 一 二 二 二 二 二 二 二 二 二 二 二 二 二
<ul> <li>◆ 安装</li> <li>第定设备的测试未通过 能探索</li> <li>※测试过程的关键截图</li> <li>S测试过程的关键截图</li> <li>S削试过程的关键截图</li> <li>S削试理和</li> <li>S削试理和</li> <li>S削试理和</li> <li>S削减用</li> <li>S削</li> <li>S削</li> <li>S</li> <li>S</li></ul>	● 启动         ● 启动         注,単击远程真机按结         日/视频,和异常汇总         ○ //视频,和异常汇总         ○ // (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)         ● (1)	<ul> <li>(未通过)</li> <li>(未通过)</li> <li>Ripper智能探索</li> <li>田,申请使用相同机型</li> <li>10 V = 4 0 0000</li> <li>2003年</li>     &lt;</ul>	<ul> <li>● 卸载</li> <li>型设备,进行测试结果</li> <li></li> <li>&lt;</li></ul>	د بیک (۱۹۹۵) بیک (۱۹۹۵) بیک (۱۹۹۵) بیک (۱۹۹۵) بیک (۱۹۹۵)		· · · · · · · · · · · · · · · · · · ·

在查看截图区域,单击设备日志按钮,查看/下载指定设备的日志详情。

序号	错误类型 💿 名称	总次数 🍦	操作
1	可忽略异常 java.lang.ClassNotFoundException: com.alibaba.triver.tools.ToolsManifest	10	详情 へ
<mark>〇</mark> 通过	[Class.forName(java.lang.String)等方式试图通过String找到类而未找到的报错,这种错误只能在运行期抛出,请确认类路	径正确并已正确引用	月。详细说明
java	.lang.ClassNotFoundException: com.alibaba.triver.tools.ToolsManifest		
	at java.lang.Class.classForName(Native Method)		
(	at java.lang.Class.forName(Class.java:309)		
(	at java.lang.Class.forName(Class.java:273)		
	at com.alibaba.triver.TriverManifest.getManifest(Taobao:912)		
(	at com.alibaba.triver.TriverManifest.getProxies(Taobao:517)		
(	at com.alibaba.ariver.AriverManifest.getProxies(Taobao:86)		
(	at com.alibaba.ariver.integration.d.getProxies(Taobao:69)		
(	at com.alibaba.ariver.integration.RVInitializer.setupProxy(Taobao:236)		
(	at com.alibaba.ariver.integration.RVInitializer.init(Taobao:101)		
(	at com.alibaba.ariver.integration.RVInitializer.init(Taobao:87)		
(	at com.alibaba.triver.c.a(Taobao:89)		
(	at com.alibaba.triver.c.a(Taobao:153)		
(	at tb.bqt.b(Taobao:191)		
(	at tb.bqt.a(Taobao:68)		
(	at com.taobao.android.nav.Nav.to(Taobao:770)		
(	at com.taobao.android.nav.Nav.to(laobao:741)		
(	at com.taobao.android.nav.Nav.intentforUri(laobao:5//)		
(	at com, taobao, android, nav. Nav. touri (Taobao, 408)		
(	at com.taobao.anarola.nav.Nav.touri(laobao:362)		
(	at com.taobao.android.nome.component.event.b.a(laobao:80)		

在**异常汇总**列表,单击**操作**列的**详情**按钮,展开指定错误的详细信息,包括错误日志、问题分析和解决方案。

#### 性能分析

查看应用运行过程中,CPU/内存/网络资源占用、运行情况的统计数据和时间趋势,以及细化至进程粒度的资源占用情况。





网络耗用(KB/min)

丢帧率(%)

将鼠标移动至时间趋势图上方,查看各统计项的即时数值。

进程详情				
进程	平均CPU占用率	最大CPU占用率	CPU时间(ms)	最大内存占用(MB)
com.taobao.taobao	1.74%	3.79%	4840	159.87
/data/app/com.tao	0%	0%	0	40.31
/data/app/com.tao	0%	0%	0	37.02
/data/app/com.tao	0.27%	2.3%	756	95.61
/data/app/com.tao	0%	0.04%	3	40.46
com.taobao.taobao	7.54%	38.09%	1170	246.68
com.taobao.taobao	1.78%	4.07%	4457	219.2
com.taobao.taobao	0%	0%	0	139.55
/data/app/com.tao	0%	0%	0	40.47

## 错误日志

在**错误日志**页签:

- 1:按照**崩溃/卡死/可忽略异常**的分类,查看错误日志。
- 2:错误日志默认显示部分内容,单击扩展按钮,查看完整日志。

3:单击**下载错误日志**按钮,下载指定的错误日志。

## 4: 单击关联设备按钮,查看引发当前错误的设备;从下拉列表中单击设备名称,打开指定设备的设备详情页面。

概况	设备详情	错误日志	1 报告	全部截图			
全部类型	崩溃(	0)	卡死 (0)	可忽略异常 (6		3	4
可忽略昇 java.io at at at at at	学常 (共 4 次) FileNotFound Libcore.io.Ic java.io.FileJ android.app.C android.conte com.mcto.ads. com.mcto.ads.	i java.io.f Exception: Bridge.oper InputStream ContextImpl ent.ContextI a.f.b.b(Uni a.f.b.b(Uni	FileNotFoundE /data/user/0 n(IoBridge.ja . <init>(FileInpu: openFileInpu: Wrapper.openF known Source: known Source:</init>	Exception: /da /com.qiyi.vid va:496) nputStream.ja t(ContextImpl ileInput(Cont 15) 67)	ta/user/0/com.qiyi.video/files/cupidBootScreenResponse: open failed: ENOENT (No such file or directory) leo/files/cupidBootScreenResponse: open failed: ENOENT (No such file or directory) va:159) .jova:612) extMrapper.java:206)	下载错误日	日志 关联设备 🗸
					2		

### 性能报告

在**性能报告**页签,查看**性能概况**和**机型性能**。

区域	说明				
性能概况	汇总统计的性能数据。				
机型性能	按照单机型分别统计的性能数据。				
概况 设备详情 错误日志 性能报告 性能概况	15 截图				

治理開催が 冷启动耗时(ms) 524	丢帧率(%) 6.94	)	CPU平均占用(%) <b>3</b> .33	内存平均占用(MB) 443.05	流量耗用(KB/min <b>46</b> .64	) 电量耗炉 <b>3</b> .59	刊(mAh/min)
机型性能							
终端设备	启动时长(ms) 🍦	平均FPS 🍦	CPU平均占用率(%) 🝦	内存平均占用率(MB) 🍦	流量耗用(KB/min) 🝦	电量耗用(mAh/min) 👙	操作
—加 GM1910 10	524	40.17	3.34	443.05	46.64	3.6	2 详情 报障 3

在**机型性能**区域,单击操作列的详情链接,打开指定设备的详情。具体内容请参见:设备详情。

如测试结果与预期不符,或其他异常情况,单击**操作**列的**报障**按钮,打开**测试任务报障**对话框,填写并提交相关信息。

测试任务报障	x
测试任务名称: No.16862 - 夏奇艺V11.9.5(8001109	165) Android編世世派式
报簿设备:	
(问题类型:	
<ul> <li>机型异常</li> <li>网络异常</li> <li>弾</li> <li>其性:</li> </ul>	順中断 ○ 自动化异苯中断
	<b>82</b> 838
└─────	
概 况 设条详情	供逗口去 性能捉在 全紙範囲 【
2	
▽ 筛选	<b>6</b>
分类方式:	
品牌	
小米	< <b>4</b>
TerMe 1.0	
小米 MI 6 8.0.0	
	共1张图片 〈 上一页 1 下一页 🏳
在 <b>全部截图</b> 页签:	
1:单击 <b>筛选</b> 按钮,	设置 <b>分类方式</b> 及其取值,查看指定分类的截图。
2: 单击右侧大图//	<b>小图</b> 切换按钮,按照指定方式显示截图。
5.2. 功能	测试
5.2.1. 提到	交测试
前提条件	
● 已购买基础测试	资源包,或使用量未超过免费阈值。具体说明及操作参见: <mark>产品定价</mark> 。
● 已上传待测试的A	Android应用。具体操作参见: <del>上传应用</del> 。
<ul> <li>已创建/导入功能</li> </ul>	测试的用例集。具体操作参见:新建/号入用例集。
<ul> <li>已上传/在线录制</li> <li>百打开移动测试器</li> </ul>	脚本,开创建用例。具体操作参见:新建用例。 ^{众制} 公五页 具体操作会见:打开放制公页页
<b>运行</b> 行业 网络武学	INIXW。来他床下罗龙·17/IINIXW。
1: 在移动测试控制	台的左侧导航栏,洗择 <b>测试产品 &gt; 功能测试</b> 页签。
2:在应用下拉列表	·····································
②谐阻	
待测试的Android	应用需预先上传;如未上传,单击 <b>上传应用</b> 按钮,即刻上传。具体操作参见: <mark>上传应用</mark> 。
2. 英士坦大测计	
3: 平古提父测试,	
移动测试	MURDIN   1/1000 W
→ 規定 → 規定 → 規定 → 規定 → 単	
功能测试 2	概 览 测试报告
性能测试 New	应用版本: V9.15.0(304)(2020-11-19 18:46:43) >
测试管理 ~	
解决方案 🗸	
> ^{计费相关} 帮助	
	暂无测试报告 调点击组交测试马上进行振测

移动测试

4:在运行功能测试页面,按步骤提示,对 5:配置完成后,将鼠标移动至提交测试,	测试进行配置。具体内容参见: <mark>配置说明</mark> 。 显示当前测试预计消耗资源包。	
預计消耗资源包 294 台次 >>>购买资源包 提交测试成功即消耗资源包,无法退回或取消 上一步 提交测试 取消		
⑦ 说明 预计消耗资源包如超过当前账户的免费	閱值,或超过已购资源包限额,单击 <b>购买资源包</b> 链接,购买 <b>基础测试</b> 资源包。	
6:单击 <b>提交测试</b> ,提交测试。		
→ 选择应用及版本 ————————————————————————————————————	测试配置	3 完成
测试提交后 排队等待执行,测试执行完毕	✔ 提交测试成功 平台当前还有 5 个任务等待执行,请耐心等待,执行完毕后可查看完整测 试报告。 返回功能测试	
	/////////////////////////////////////	
选择应用及版本		
ASGATINE     KODA     NU     NU	○ maxe	
Ted Ex		
选项	说明	
应用包	默认选中已上传/选择的应用,可修改。	
	默认选中已上传/选择的应用的版本。如应用存在多个版本,则选择用于功能测试的版本。	
应用版本	② 说明 若待测试的应用为混淆版本,请先上传符号表,应用解析成功后再提交测试。上传符号表的具体操作请参见上传符号表。	
测试备注	输入测试相关信息,便于后续查询回顾。	
<b>测试配置</b> 基础信息 正式版 体验版		
服务项 <ul> <li>自定义测试机型</li> <li>用例测试结果</li> <li>Crash/ANR检测</li> <li>账号池</li> </ul>	<ul> <li></li></ul>	
送择机型 已选择 10 款		

## **基础信息**分组包括:

	选项	说明
--	----	----

选项	说明			
正式版/体验版	<ul> <li>正式版:已购买基础测试资源包,付费使用的版本。</li> <li>⑦ 说明 使用正式版运行功能测试,需购买基础测试资源包。</li> <li>体验版:使用量在免费阈值内,体验使用的版本。</li> <li>⑦ 说明 在体验版的计费项列,可查看当前账号的免费额度;测试超出的使用量,需按量付费。</li> </ul>			
选择机型	<ul> <li>1: 单击选择机型,打开选择提测机型右侧栏,手动指定或通过预选项批量指定预选机型。</li> <li>ZHURNEL</li> <li>IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</li></ul>			

#### 测试配置

* 用例库版本:			
1.0.0			
★ 测试用例集 :			
选择用例集	用例管理		
用例集名称	描述	所属目录	操作
WLD用例集	文档开发	WLDTEST	移除
參数分组:			
默认参数分组			
胀号互踢:			
<b>香</b>			
邮件通知地址:			
邮件通知地址			

#### **测试配置**分组包括:

选项	说明
用例库版本	指定功能测试用例集所在的用例版本库。
测试用例集	指定功能测试用例集。 ⑦ 说明 功能测试用例集需预先设置;如未预先设置,单击用例管理按钮,打开测试管理>脚本管理页签,即刻进行设置。具体操作参见:新建/导入用 例集。

选项	说明
参数分组	指定功能测试用例集相关的参数分组。 ⑦ 说明 登录用例集的相关参数分组需预先设置,具体操作参见:新建参数分组。
账号互踢	设置应用账号是否仅限在唯一终端登录。 ⑦ 说明 账号互踢设置为是时,需在用例管理页面的参数池页签中配置参数account,以添加相关账号。配置参数的具体操作请参见新建/导入参数。
邮件通知地址	设置电子邮箱地址,用于接收测试完成通知。

#### 后续操作

查看概览信息

查看测试报告

## 5.2.2. 查看概览信息

### 前提条件

- 已提交功能测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

### 操作步骤

- 1: 在移动测试控制台的左侧导航栏,选择测试产品 > 功能测试页签。
- 2: 在应用下拉列表选择已提交测试的Android应用,默认打开概览页面。

移动研发平台EMAS / 移动测试 / 功能测试



#### 3:在**概览**页面:

- 从应用版本下拉列表中选择应用的版本;
- 从提测于下拉列表中,选择提测时间,默认为最近一次提测;
- 查看指定应用/版本/提测时间的测试概览信息。具体内容参见:概览页面说明
- 4: 单击查看报告按钮, 打开指定测试的详情页面。具体内容参见: 测试报告说明
- 5: 如当前测试状态为运行中,单击刷新按钮,手动刷新概览页面,查看最新信息。
- 6:单击**下载报告**按钮,下载指定测试的测试报告。

#### ? 说明

如当前测试状态为**运行中**,则**下载报告**按钮非活性。

## 概览页面说明



#### Android功能测试概览页面包括:

信息	说明
测试状态	显示当前测试的状态,包括 <b>运行中</b> 和 <b>已完成</b> 。
用例结果	显示当前用例中运行中/失败/未执行/取消的设备数,及所占比例。
错误分布	显示当前测试中发生的错误类型、影响设备数,及各类错误所占比例。
测试结果	基于每测试设备,显示当前测试的测试结果。

## 5.2.3. 查看测试报告

## 前提条件

- 已完成功能测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面

## 查看测试报告

- 1:在移动测试控制台的左侧导航栏,选择**测试产品 >功能测试**页签。
- 2:在应用下拉列表选择已测试的Android应用,选择**测试报告**页签,查看指定应用执行的功能测试列表。

移动研发平台EMAS / 移	<b>动测试 / 功能测试</b>						
移动测试	3						
测试产品 1 ^	📥 手机淘宝	> ▷ 提交测	□试 ① 上传应用				
兼容測试	概览测试报	告 4					
功能测试 2							
远程真机							
测试管理 🖌 🖌	编号	应用版本	提测时间	測试结果 🛛	状态	〒 备注	操作
解决方案 >	719067	9.15.0	2020-11-26 17:39:00	10 (0/10/0)	● 已完成	WLDTEST	⊿ 查看 分享 6
计费相关							〈 上一页 】 下一页 〉
帮助							
• 如测试正在运行 • 如测试正在运行 5.2.4.测	中, 单击取消按钮 试报告说	11.111日11月2日日日 11.11日日日日日 明	<b>西</b> 厄刀子吐孜, 正忌用厂	<b>圴</b> 世用 <b>风见</b> 奋重4	目以別以口。		
概述							
移动研发平台EMAS / 移	多动测试 / 任务管理 / 测词	式报告详情					
← 手机淘	宝(V9.15.0(	304)) 💿					日完成 下載   分享 2
概 况 设备详情	青 错误日志 性的	能报告 全部截图 1					

#### 1:选择各页签查看相关内容。

页签	说明
概况	查看当前测试的执行结果、错误日志、用例集概况。
设备详情	基于每测试设备,查看/编辑测试用例的执行结果,及相关日志。
错误日志	查看当前测试的报错信息,包括 <b>崩溃、卡死</b> 和和其他 <b>可忽略异常</b> 。
性能报告	查看当前测试的应用性能统计数据,以及,基于每测试设备的性能数据。
全部截图	查看当前测试的关键截图。

#### 2: 单击右上角各按钮,进行相关操作。

按钮	说明
下载	下载当前测试报告。
分享	生成当前测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。

#### 3: 鼠标移动至帮助按钮, 查看当前测试基本信息。

ID: 719067
測试类型:Android功能测试
提测人: e*********r
提测时间:2020-11-26 17:39:00
測试账号互踢: 否

## 概况

## 在概况页签,查看当前测试的执行结果、错误日志、用例集概况。

## 执行结果

## 显示通过/未通过/异常未执行的机型数及其占比。



## 错误日志

#### 显示关键错误的列表。

错误日志			
错误类型	出现次数	关键错误	操作
卡死	2	0,14256,com.taobao.taobao,9	详情
卡死	1	0,9394,com.taobao.taobao,12	详情
卡死	1	0,28710,com.taobao.taobao,1	详情
可忽略异常	21	java.lang.ClassNotFoundExce	详情
□ 查看全部			

单击**查看全部**按钮,或者单击**详情**按钮,打开**错误日志**页签,查看全部错误及其日志详情。具体内容参见:错误日志

#### 用例集概况

基于每用例集,显示当前应用的测试结果。

用例集概况					
用例集名称	用例库版本	目录	用例集类型	成功/失败/异常(机型)	±1 2
WLD用例集	1.0.0	WLDTEST	录制用例	3 / 5 / 2	用例概况 > 机型概况 >

1:在用例集列表,单击操作列的用例概况按钮,显示用例集用例的详细信息。

用例集構	既况							
	用例集名称      用例库版本		目录		成功/失败/异常(机型)		操作	
	WLD用例集	D用例集 1.0.0		WLDTEST 录制用例 3		5/2	用例概况 ヘ   机型概况 >	
	用例名称		用例描述			脚本名称	操作	
	WLDTEST在线录制用例		用户指南文档开发测试用例		WLDTEST在线录制脚本		查看	

### 2: 单击机型概况按钮,基于每测试设备,显示测试结果。

概况							
用例集名称	用例集名称 用例库版本 目录		用例集类型	成功/失败/异常(机型)	功/失败/异常(机型)		
WLD用例集	1.0.0	WLDTEST	录制用例	3 / 5 / 2		用例概况 >   机型概况 >	
设备名称			状态	安装	执行	操作	
vivo V1813T(Y97) 8.	.1.0		未通过	0	8	执行日志 🗌 设备详情	
vivo V1938T(x30 pro	o) 9		通过	0	۲	执行日志 🗌 设备详情	
华为 VTR-AL00 (P10)	) 7.0		未通过	0	8	执行日志 🗌 设备详情	
华为 MED-TL00(畅享	10e) 10		通过	٥	٢	执行日志 设备详情	
金立 S10 7.0			未通过	٥	8	执行日志 设备详情	
vivo V1732A(Y81s) 8	8.1.0		通过	٥	٥	执行日志 设备详情	
联想 Lenovo L38111	9		异常/未执行	٥	0	执行日志 设备详情	
糖果 F9 6.0.1			异常/未执行	٥	0	执行日志 \ 设备详情	
中兴 ZTE Q301C 4.3			未通过	0	8	执行日志 \ 设备详情	
—加 GM1910(—加7F	Pro) 10		(未通过)	0	0	执行日志 🗌 设备详情	

3:在机型概况列表,单击执行日志按钮,查看指定设备的执行日志;单击设备详情按钮,打开指定设备的设备详情页面,查看详细信息。具体内容参见:设备详情 设备详情

#### хдлы

## 在**设备详情**页签:

1:在左侧**设备**列表,单击**筛选**下拉按钮,按照**分类方式**或**状态**筛选设备。

筛选项	说明
分类方式	将测试设备按照指定分类方式进行分组,包括 <b>按测试结果、按系统版本、按品牌</b> 3个选项。
状态	将测试设备按照指定状态进行筛选,包括 <b>全部、未通过、设备错误、异常/未执行、超时、通过、等待中、运行中</b> 。 当 <b>状态</b> 设置为 <b>未通过</b> 选项,可进一步按照 <b>失败环节</b> 进行筛选,包括 <b>安装、启动、运行、卸载</b> 4个环节。

## 2: 在左侧设备列表,单击设备的名称,在右侧打开指定设备的详细信息。

後 🎫 🛛 🛛	∀ 筛选	中兴 - ZTE Q3010	2-4.3 (异常/未现行)		∅ 远程真
/ 未通过	分类方式 ·	<ul> <li>rhit</li> </ul>	SWLD用例集		
vivo - V1813T(Y97) - 8.1	按测试结果(	(默认) 🗸	WLD用例集 V		
—加 - GM1910(—加7Pro)	状态:				
华为 - VTR-AL00 (P10) - ⁻ 金立 - S10 - 7.0	全部		00:00:04		🕞 视频 📄 脚本日志 📄 Appium日志 📄 设备日志
通过	3				
vivo - V1732A(Y81s) - 8.1	.0			暂无步骤数据	
vivo - V1938T(x30 pro) - 华为 - MED-TL00(畅享10e	9				
异常/未执行	3	异常汇总			
中兴 - ZTE Q301C - 4.3	3	序号	错误类型 🛒 名称		总次数 🖕 操作
糖果 - F9 - 6.0.1	<b>-</b> - 1				
联想 - Lenovo L38111 - 9					

#### 显示指定设备各用例集的执行状态。

 中兴 - ZTE Q301C - 4.3 (新魚洗杯)
 ● 这程燕

 ● 安菜
 ● WLD用砌集

 如指定设备的测试未通过,单击远程真机按钮,申请使用相同机型设备,进行测试结果验证.

 *D***约集执行结果** 

 1 在查看用例集执行结果下拉列表,选择用例集,查看指定用例集的执行概况.

 2: 单击视频按钮, 查看指定用例集的执行视频记录;单击脚本/Appium/设备日志按钮,查看/下载指定用例集的各类执行日志.

 查看用例集执行结果:
 ● WLD用例集

 ● WLD用例集
 ●

 ● WLD用
 ●

 ● ●
 ●

 ●
 ●

 ●
 ●

 ●
 ●

 ●
 ●

 ●
 ●

### 异常汇总

查看指定用例集的异常信息。

异常汇)	总			
	序号	错误类型 💿 名称	总次数 👙	操作
	1	可忽略异常 java.lang.ClassNotFoundException: com.alibaba.triver.tools.ToolsManifest	10	详情 へ
	♀ 通过C	lass.forName(java.lang.String)等方式试图通过String找到类而未找到的报错,这种错误只能在运行期抛出,请确认类路径正码	确并已正确引用	。详细说明
	java.l	ang.ClassNotFoundException: com.alibaba.triver.tools.ToolsManifest		
	at	java.lang.Class.classForName(Native Method)		
	at	java.lang.Class.forName(Class.java:309)		
	at	java.lang.Class.forName(Class.java:273)		
	at	<pre>com.alibaba.triver.TriverManifest.getManifest(Taobao:912)</pre>		
	at	<pre>com.alibaba.triver.TriverManifest.getProxies(Taobao:517)</pre>		
	at	<pre>com.alibaba.ariver.AriverManifest.getProxies(Taobao:86)</pre>		
	at	com.alibaba.ariver.integration.d.getProxies(Taobao:69)		
	at	com.alibaba.ariver.integration.RVInitializer.setupProxy(Taobao:236)		
	at	com.alibaba.ariver.integration.RVInitializer.init(Taobao:101)		
	at	<pre>com.alibaba.ariver.integration.RVInitializer.init(Taobao:87)</pre>		
	at	com.alibaba.triver.c.a(Taobao:89)		
	at	com.alibaba.triver.c.a(Taobao:153)		
	at	tb.bqt.b(Taobao:191)		
	at	tb.bqt.a(Taobao:68)		
	at	com.taobao.android.nav.Nav.to(Taobao:770)		
	at	com.taobao.android.nav.Nav.to(Taobao:741)		
	at	com.taobao.android.nav.Nav.intentForUri(Taobao:577)		
	at	com.taobao.android.nav.Nav.toUri(Taobao:408)		
	at	com.taobao.android.nav.Nav.toUri(Taobao:362)		
	at	<pre>com.taobao.android.home.component.event.b.a(Taobao:80)</pre>		
	at	<pre>com.taobao.android.home.component.event.b.a(Taobao:62)</pre>		
	at	<pre>com.taobao.android.home.component.event.b.a(Taobao:30)</pre>		
	at	tb.iuz.handleEvent(Taobao:81)		

在**异常汇总**列表,单击操作列的详情按钮,展开指定错误的详细信息,包括错误日志、问题分析和解决方案。

解决方案	×
问题分析:	
ClassNotFoundException是通过Class.forName(java.lan g.String)等方式试图通过String找到类而未找到的报错,这种错 设印能在设行期端出。	
解决方案:	
通过Class.forName(java.lang.String)等方式试图通过Stri ng找到类而未找到的报错,这种错误只能在运行期驰出,请确认类 路径正确并已正确引用。	
代码示例:	
<pre>package com.alibaba.mqc.test;</pre>	
<pre>import java.lang.reflect.InvocationTargetE xception;</pre>	
<pre>import java.lang.reflect.Method;</pre>	
<pre>//简单的反射demo public class V {     public void printf(){         System.out.println("printf");     } }</pre>	**
public static void main(String] args) throws ClassNotFoundException, SecurityException, NoSuchMethodException, IllegalArgumentException, I llegalAccessException, InvocationTargetException,	

#### 性能分析曲线

查看指定用例集在运行过程中,CPU/内存/网络资源占用、丢帧率和电量随时间变化的趋势。

1		
0	11-26 17:42:17	
	• CPU: 0 %	
	• 内存: 0 MB   PSS: 0 MB HeapAlloc: 0 MB	
0	● 网络: 918 KB 上行: 0 KB 下行: 918 KB	,
1000	● 丢帧率: 0%	
/	● 电量: 0.2 mA   CPU: 0 mA WIFi: 0.1 mA 屏幕: 0.1 mA	
500		
0		
1		
0		
0.24		

## 将鼠标移动至时间趋势图上方,查看各统计项的即时数值。

## 错误日志

## 在**错误日志**页签:

- 1:按照**崩溃/卡死/可忽略异常**的分类,查看错误日志。
- 2:错误日志默认显示部分内容,单击扩展按钮,查看完整日志。
- 3: 单击下载错误日志按钮, 下载指定的错误日志。
- 4: 单击管理设备按钮, 查看引发当前错误的设备。

概况	设备详情	错误日志	1 报告	全部截图													
全部类型	崩溃	(0) -	卡死 (0)	可忽略异常 (6	)										3	4	
可忽略 java.io at at at at at	早常 (共 4 次 .FileNotFoun libcore.io.I java.io.File android.app. android.cont com.mcto.ads com.mcto.ads	) : java.io.F dException: oBridge.open InputStream. ContextImpl. ent.ContextW .a.f.b.b(Unk .a.f.b.b(Unk	ileNotFoundE /data/user/0/ ([IoBridge.jav <init>(FileIr openFileInput Irapper.openFi mown Source:1 mown Source:6</init>	<pre>ixception:/da /com.qiyi.vid /a:496) nputStream.ja t(ContextImpl ileInput(Cont 15) 67)</pre>	ta/user/O/com.qiy leo/files/cupidBoo va:159) .java:612) extWrapper.java:2	i.video/files/c otScreenRespo 206)	/cupidBoot	ntScreenRe	esponse: ENOENT (	open faile No such f	ed:ENOE file or d	ENT (No su directory)	ıch file or di	irectory)	下载错误	日志	设备 ∨
							V	v 2									

## 性能报告

### 在**性能报告**页签,查看**性能概况**和**机型性能**。

区域	说明
性能概况	汇总统计的性能数据。
机型性能	按照单机型分别统计的性能数据。

能概况	-							
冷启动耗时(ms) 894	丢帧率(%) <b>62</b> .50	c (	PU平均占用(%) <b>〕</b> .31	内存平均占用(MB) <b>38</b> .22	流量耗用(KB/min) <b>43</b> .60	电量耗用(mA 0.71	sh/min)	
1型性能								
终端设备	启动时长(ms) 🍦	平均FPS 🍦	CPU平均占用率(%) 🍦	内存平均占用率(MB) 🍦	流量耗用(KB/min) 🝦	电量耗用(mAh/min) 🝦	操作	
vivo V1813T(Y97) 8.1.0	1337	2	0.36	53.94	22.27	1.13 2	详情	报障
vivo V1938T(x30 pro) 9	481	4	0.15	65.55	65.36	0.45	详情	报障
华为 VTR-AL00 (P10) 7.0	561	1	0.23	34.12	0	0	详情	报障
华为 MED-TL00(畅享10e) 10	1046	2	0	34.27	124.54	0	详情	报障
金立 S10 7.0	797	0	0	38.97	25.04	0	详情	报障
vivo V1732A(Y81s) 8.1.0	1796	2	1.36	35.19	155.9	0.83	详情	报障
联想 Lenovo L38111 9	1014	1	1.06	61.95	19.2	2.01	详情	报障

## 在**机型性能**区域,单击操作列的详情链接,查看指定设备的详情。具体内容参见:设备详情

如测试结果与预期不符,或其他异常情况,单击**操作**列的**报障**按钮,打开**测试任务报障**对话框,填写并提交相关信息。

测试任务报障	×
测试任务名称:	
No.16862 - 爱奇艺V11.9.5(800110955) Android兼容性测试	
报降设备:	
问题类型:	
<ul> <li>机型异常</li> <li>网络异常</li> <li>弹窗中断</li> <li>自动化异常中断</li> </ul>	
其他:	
線交 取2	4
報交 取3	Ħ

## 全部截图

	概 况	设备详情	错误日志	性能报告	全部截图	1							
	2												
	日第	选									e		
	分类7	5式:											
	品版	₽ ∠	×	3									
>	10		v	4									
		TestMe 1.8 R mit Ricce Mittany											
	小米	MI 6 8.0.0											
										共 1 张图片	< 上一页	1 下一	

在**全部截图**页签:

1: 单击筛选按钮,设置分类方式及其取值,查看指定分类的截图。

2:单击右侧**大图/小图**切换按钮,按照指定方式显示截图。

# 5.3. 性能测试

# 5.3.1. 管理测试用例

## 前提条件

已打开移动测试控制台页面。具体操作参见:打开控制台页面。

#### 打开性能测试页面

在移动测试控制台,选择**测试产品 > 性能测试**,打开**性能测试**页面。

12

移动研发平台EMAS	/ 移动	则试 / 性能测试	
移动测证	Ĵ		
测试产品 1	^	🛎 手机淘宝 🗸 🗸	□ 运行激试
兼容測试		用例列表	
性能测试 New	] @		
远程真机			选择用例查看
测试管理	~	暂无用例数据	您可选择用例查看或者直接运行测试
解决方案	~	请新建用例	运行测试
> 计费相关		新建用例 新建目录	
帮助			

## 新建目录

1:在性能测试页面,如用例列表为空,即初次创建目录,单击新建目录;或从用例列表的扩展菜单中,选择新建目录选项,打开新建目录对话框。

📥 手机淘宝	∨ 区 运行测试	
用例列表	: 2	
	新建用例	
	新建目录 3	选择用例查看
暂无用例	数据	您可选择用例查看或者直接运行测试
请新建月	目例	运行测试
新建用例	新建目录 1	

### 2:在新建目录对话框,填写目录信息。

属性	说明
父级目录	用于指定目录的存储路径,即所在父级目录。
目录名称	用于设置目录的名称。 【取值范围】2~50字符 【字符类型】中文/英文大小写/数字/下划线/短横线
	↓ 注意 同级目录的名称不能相同。

#### 3:单击**确定**,保存目录。

目录创建后,返回**性能测试**页面,在**用例列表**查看已创建的目录。

📥 手机淘宝		🗵 运行测试
用例列表	:	
▼ 🖻 WLDTEST目录		

## 新建用例

1:在性能测试页面,如用例列表为空,即初次创建用例,单击新建用例;或从用例列表的扩展菜单中,选择新建用例选项,打开新建用例对话框。

📥 手机淘宝	~	と 运行測试		
用例列表	-	2		
	新建用例	3		
	新建目录		选择用例查看	<u>-</u>
暂无用例数	如据		您可选择用例查看或者直接	<b>妾运行测试</b>
请新建用例	U		运行测试	
1 新建用例 新建	建目录			

## 2:在新建用例对话框,填写用例信息。

属性	说明
父级目录	用于指定用例的存储路径,即所在父级目录。
用例名称	用于设置用例的名称。 【取值范围】2~50字符 【字符类型】中文/英文大小写/数字/下划线/短横线

#### 3:单击**确定**,保存用例信息。

用例创建后,返回**性能测试**页面,在**用例列表**查看已创建的用例;单击已创建的用例,在右侧显示用例信息。

🛎 手机淘宝	$\vee$	🗵 运行测试							
用例列表	:	WLDTE	ST用例						
▼		任务列	長 用例详情	i					
WLDTEST用例	I	应用版本:	9.15.0(304)	$\vee$	选择设备: 全部	~			
			CPU占用		内存占用		帧率	网络均值	
			0.00 %		Омв		O FPS	O кв/s	
		名称	开始时间	测试时长	CPU占用(%)	内存占用(MB)	帧率(FPS)	网络均值(KB/S)	操作
						暂无数据			
									操作
								< 上一页	1 下一页 >

? 说明

将鼠标滑动至**用例列表**中已创建的目录上,从快捷菜单中选择**新建用例**选项,在指定目录下创建用例。

## 管理用例列表

1: 将鼠标移动到已创建的目录上, 单击快捷按钮, 对目录进行管理。



## 移动测试

选项	子项	说明				
æ	新建用例	在当前目录下,新建用例。				
	新建目录	在当前目录下,新建子目录。				
2	-	重命名当前目录。				
		删除当前目录。				
X	-	注意 删除目录后,目录下所有用例将被同时删除,且数据不可恢复,请谨慎操作。				

## 2: 将鼠标移动到已创建的用例上, 单击快捷按钮, 对用例进行管理。

用例列表	
➤ 卧 WLDTEST目录 ① WLDTEST用例	
选项	说明
<u> </u>	重命名当前用例。
	删除当前用例。
X	✓ 注意 删除用例后,数据将不可恢复,请谨慎操作。

3:选中已创建的目录/用例,长按鼠标左键,并将目录/用例拖动到适当位置。

## 后续操作

提交测试 查看测试报告

## 5.3.2. 提交测试

### 前提条件

- 已购买远程真机资源包,或使用量未超过免费阈值。具体说明及操作参见:产品定价。
- (可选)已上传待测试的Android应用。具体操作参见:上传应用。
- (可选)已创建测试目录和用例。具体操作参见:管理测试用例。
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面。

#### 操作步骤

- 1:在移动测试控制台的左侧导航栏,选择**测试产品 > 性能测试**页签。
- 2:在应用下拉列表选择待测试的Android应用。

#### ? 说明

待测试的Android应用需预先上传,具体操作参见:<mark>上传应用</mark>;如未上传,待后续操作过程中上传。

#### 3:在用例列表选择已创建的用例。

```
    ⑦ 说明
    测试目录和用例需预先设置,具体操作参见:管理测试用例;如未设置,待后续操作过程中设置。
```

## 4:单击**运行测试**,打开**远程真机**页面。

移动研发平台EMAS / 移动测试 / 性能测试							
移动测试 🛛 🔞							
测试产品 1 ▲ 手机淘宝	∨ □ 运行测试	5					
兼容测试 功能测试 <b>用例列表</b>	: WLDTE	EST用例					
性能测试 🚥 🝳 🕶 🗠 WLDTEST	任务列	表 用例详情					
过程真机 □ WLDTI 测试管理	EST用例 4 应用版本:	全部 🗸	选择设备: 全部				
解决方案 🖌		CPU占用	内存占用	帧率		网络均值	
> 计费相关		0.00 %	Омв	0 F	PS	O кв/s	
帮助	名称	开始时间 测试时长	CPU占用(%)	内存占用(MB)	帧率(FPS)	网络均值(KB/S)	操作
				暂无数据			

#### 5:在远程真机页面:

- 申请运行性能测试的设备,打开指定设备的远程真机页面。具体操作参见:申请远程真机。
- 在指定设备的远程真机页面 > 应用页签,安装待测试的Android应用。具体操作参见:安装应用。
- 在指定设备的远程真机页面 > 性能测试页签,对Android应用进行性能测试。具体操作参见:实施性能测试。

### 后续操作

查看测试报告

## 5.3.3. 查看测试报告

## 前提条件

- 已申请远程真机实施性能测试,并将测试数据保存至指定用例。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

### 打开用例页面

- 1:在移动测试控制台,打开**测试产品 > 性能测试**页面。
- 2:选择已进行性能测试的应用,并在用例列表中选择用例,在右侧打开指定用例页面。

移动研发平台E	MAS / 移詞	测试 / 性能测试								
移动测	试	3								
测试产品	1 ^	🛎 手机淘宝 🛛 🗸	🗵 运行测试							
兼容测试 功能测试		用例列表	WLDTEST用例							5
性能测试	New 2	▼	任务列表 用例详情							
远程真机 测试管理	•	C WLDTEST用例 4	应用版本: 9.15.0(304)	∨ 选择设备: 全部						
解决方案	~		CPU占用	内存占用		帧率			网络均值	
> 计费相关			8.76%	52.59 мв		<b>4</b> .58 F	PS		286.34 к	B/S
帮助			名称	开始时间	测试时 长	CPU占用 (%)	内存占用 (MB)	帧率 (FPS)	网络均值(KB/S)	操作
			2020-11-30 14:29:27 手机淘宝 (com.taobao.taobao)	2020-11-30 14:29:29	01:10	8.76	52.59	4.58	286.34	详情 删除
									< 上一页	1 下一页
			L							50

## 查看任务列表

#### 操作说明

- 1:在指定用例页面,打开**任务列表**子页签。
- 2: 输入筛选条件,查询满足条件的测试任务,及其统计信息。
- 3:在任务列表中,单击操作列的详情按钮,打开指定任务的任务详情页面。
- 4: 单击操作列的删除按钮, 删除指定任务。

WLDTEST用例							
任务列表用例详情	2						
应用版本: 9.15.0(304)	✓ 选择设备: 全部		$\sim$				
CPU占用 <b>8</b> .76 %	内存占用 <b>52</b> .59 MB		帧率 <b>4</b> .58 F	PS		网络均值 <b>286</b> .34 KI	B/S
名称	开始时间	测试时 长	CPU占用 (%)	内存占用 (MB)	帧率 (FPS)	网络均值(KB/S)	操作
2020-11-30 14:29:27 手机淘宝 (com.taobao.taobao)	2020-11-30 14:29:29	01:10	8.76	52.59	4.58	286.34	详情 删除
						< 上一页	1 下一页

#### 筛选条件

选项	说明
应用版本	用于指定应用的版本。
选择设备	用于指定测试设备的品牌型号。默认为 <b>全部</b> 。
统计数据	
选项	说明

CPU占用	指定应用版本在指定测试设备上的平均CPU占用率。
内存占用	指定应用版本在指定测试设备上的平均内存占用率。
帧率	指定应用版本在指定测试设备上的平均帧率。
网络均值	指定应用版本在指定测试设备上的平均网络传输速率。

## 查看用例详情

操作说明

1:在指定用例页面,打开**用例详情**子页签。

2: 输入筛选条件,查询满足条件的用例及其详情。

WLDTEST用例	
-----------	--

本: 9.15	.0(304) ×	选择设备: 全部		∨ 排序:	按版本 🗸	
2 本对比		3			4	
版本	平均应用CPU(%)	平均内征	字(MB)	平均FPS	S 平均网络(KB/S)	
ALL	10.75	48.03		17.29	368.1	
9.15.0	10.75	48.03		17.29	368.1	
PU <b>趋势</b>						
12 - 10 -			0	o		
0						

9.15.0

#### 筛选条件

选项	说明
应用版本	用于指定应用的版本。可以选择1个或多个应用版本,进行版本对比。
选择设备	用于指定测试设备分组,包含 <b>高端机 / 中端机 / 低端机</b> 。默认为 <b>全部</b> 。
排序	用于设置排序的方式,包括 <b>按版本/按创建时间</b> 。默认为 <b>按版本</b> 。

## 5.4. 远程调试

#### 前提条件

- 已购买远程真机资源包,或使用量未超过免费阈值。具体说明及操作请参见:产品定价。
- (可选)已上传待调试的Android应用。具体操作参见:上传应用。
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面。

#### 申请远程真机

- 在移动测试控制台的左侧导航栏,选择测试产品 > 远程真机页签,打开远程真机资源列表。
- 2. 选择待调试的远程真机。
  - ・在搜索框输入关键字,查询指定品牌/机型的远程真机资源。
  - 单击展开筛选,展开远程真机的筛选条件,指定1个或多个选项,筛选满足条件的远程真机。

### 🥐 说明

筛选条件包括品牌/Android版本/iOS版本/分辨率/上市时间等。

#### 3: 在远程真机资源列表,将鼠标移动至设备标签,单击**立即使用**,打开远程真机页面。

? 说明

Android真机全部没有root。

	测试报告			
兼容测试				
功能测试 3 Q. 华为 展开描述 V 4 剩余体验。	时长: 30:00   剩余资源时长: 00:00 🔮			
性類的成				
远程真机 2 华为(鸿蒙) - JEF-AN● 空闲 华为 - KKG-AN70( ● 空闲 华为(鸿蒙) - ELS-AN● 空闲	<b>华为(鸿蒙) - RTE-AL…●</b> 空闲			
隐私合规 分辨率: 1080 分辨率: 1080 分辨率: 800x	分辨率: 1236			
測试管理 ^ 系统版本: 21.0 系统版本: 11 系统版本: 22.0	系统版本: 2.1.0			
田府管理 设备分级:中	设备分级:中			
应用管理				
符号表管理				
华为(鸿蒙) - WLZ-A ● 空闲 华为(鸿蒙) - ELS-AN● 空闲 华为(鸿蒙) - BRQ-A ● 空闲     解決方案     ^	<b>华为 - LLD-AL00 (荣…●</b> 空闲			
专家服务 分辨率: 720x 分辨率: 1200 分辨率: 1133	▲0 → 分辨率: 1080			
<b>专有三級务</b> 系统版本: 2.2.0 系统版本: 2.2.0 系统版本: 2.2.0	系统版本: 9			
计费相关 设备分级:中 设备分级:高 设备分级:高 设备分级:高	2018年1月11日 1月11日 1月111日 1月111日 1月111日 1月111日 1月111日 1月111日 1月111日 1月1111111 1月11111111			
	文即使用 文印使用			

## 远程真机说明

远程真机用于在Web页面显示终端设备的基本操作。



序号	按钮	控件	说明
		终端类型	远程真机的品牌、型号。
	系统设置	时长	<ul><li>远程真机的体验时长或使用资源包的时长。</li><li>远程真机的剩余体验时长和剩余资源包的时长。</li></ul>
		高清	切换设备显示屏为高清/标清显示。
		o/v 输入账号密码	在使用adb连接远程真机进行远程调试时,部分真机需进行登录验证。在这种情况下,单 击 <b>o /v 输入账号密码</b> 按钮。
		锁定屏幕	将设备锁屏。
1		旋转屏幕	切换设备横屏/竖屏显示。
		获取设备剪贴板内容	复制远程真机剪贴板上的内容到控制台。
		快捷键盘	快捷输入窗口,支持中英文、数字、普通符号输入。

DingGov 👻

序号	按钮	控件	说明
		打开应用管理	打开应用管理页面。
2	全屏按钮	-	单击 <b>全屏</b> 按钮,全屏显示远程真机。
3	系统按钮	-	用于模拟Android手机的3个系统按钮。
4	页面调节滑块	-	用鼠标拖动页面调节滑块,调整远程真机的窗口大小。

#### 安装应用

- 1. 在指定设备的**远程真机**页面,默认进入**应用管理**页签。
- 2. 查找应用。
  - 如待调试应用已上传,单击已上传/已安装,筛选相应状态的应用。
- 在搜索框输入关键字,模糊匹配应用的名称,查找指定应用。
- 3. 上传应用。如待调试应用未上传,单击**上传应用**,即刻上传,具体操作请参见:上传应用。
- 4. 安装应用。如待调测应用已上传未安装,单击**操作**列的**安装**,在远程真机上安装应用。

移动研发平台EMAS / 移动测试 / 远程真机

	-			_				
+	未插卡ඛ 含 🛊 ■ 🖗 🗮 🖸 … 🛛 🗍 第100% 🖿 傍線6:51	X 全屏	应用管理	日志	截屏 文件		更多 ∨	退出
LLD-AL00 (荣耀9青春 版)	_{余杭区} ◎ 人・ <b>「」」 へ</b> 7℃	2	日上传	已安装	◆ 上传应用	5 《 搜索	如何包 3	
③ 15:00 ②			应用	版本	文件大小	上传时间	操作	
日高清〇			东北证券 融e通	V6.1.6(202430)	131.51M	2022-01-24 15:58:46	安装 4	
-D -B			天翼云办 公	V1.0.23(1023)	28.61M	2022-01-24 10:36:39	安装	
			猎聘	V5.22.0(11960)	97.71M	2022-01-20 17:06:45	安装	
			易考过	V3.0.0.16(39)	14.44M	2022-01-19 10:09:34	安装	
<b>E</b>			东北证券 融e通	V6.1.5(202426)	131.45M	2022-01-17 13:29:48	安装	
			人人视频	V5.14.0(272)	111.54M	2022-01-17 10:10:31	安装	
	12.F		聚鲸融	V2.0.0(15)	22.52M	2022-01-11 17:26:39	安装	
			智导互联	V1.2.0.10001(1)	136.10M	2022-01-11 15:17:54	安装	
			当当云阋 读	V7.2.80(7280)	76.22M	2022-01-10 16:16:16	安装	
			东北证券 融e通	V6.1.3(211224)	128.32M	2022-01-07 14:13:42	安装	
			当当	V11.12.2(283)	120.52M	2021-12-31 15:59:00	安装	
	1991200 2001045 ages		沃钱包	V5.1.0(510)	83.79M	2021-12-30 23:46:37	安装	
			沃钱包	V5.1.0(510)	69.54M	2021-12-30 23:01:54	安装	
			风芒预发 布	V6.0.0(44)	77.60M	2021-12-30 11:45:42	安装	
			EZOpenSDk	Vv4.19.0(4190)	64.98M	2021-12-29 14:15:34	安装	
	4 0 0	J	沃钱包	V5.0.0(500)	69.26M	2021-12-29 11:04:59	安装	•
≡	 ۵	5						

-

#### 查看日志

- 1:在指定设备的**远程真机**页面,打开**日志**页签。
- 2:在日志页签,开启开/关键,开始记录设备日志。
- 3: 在选择级别下拉列表,选择日志的级别,显示指定级别的日志。
- 4: 在关键词过滤文本框,输入关键字,模糊匹配日志信息,查找满足条件的日志。
- 5:开启**滚动**开关键,日志列表滚动显示最新的日志。
- 6: 单击**下载日志**,将日志信息以文本文件的形式,下载至本地。
- 7:单击**清除日志**,清空日志列表。

						_			
	÷	未插卡D 宅 幸 🖬 😋 🖷 😰 … 🛛    🕯 後後6:56	※ 全屏	应用管理	日志	截屏	文件		更多 🗸 退出
LLD (荣耀	-AL00 四青春	¢£ic ♥	2	开 选择	级別 >	关键词过渡	l.		✓ ↓ 下载日志 清除日志
0.00				时间	PID	级别	Tag	数量	日志
0 20		1月24日星班一		18:56:30.26	2188	ERROR	adbd	1	failed to connect to socket 'loc 🔺
				18:56:30.32	17178	INFO	TrafficMonitor	1	update:rxPkts:17,txPkts:0,rxByt
Ð	₿			18:56:30.32	17178	INFO	TrafficMonitor	1	start expired. level:-1
Gà	0	O		18:56:30.32	17178	INFO	TrafficMonitor	1	gettimer:interval=2000
	99			18:56:30.37	2188	ERROR	adbd	1	failed to connect to socket 'loc
	00			18:56:30.47	2188	ERROR	adbd	1	failed to connect to socket 'loc
				18:56:30.54	589	ERROR	WifiHAL	1	enter wifi_get_link_stats
				18:56:30.55	17178	ERROR	WificondControl	1	Noise: 0, Snr: -1, Chload: -1
				18:56:30.56	17178	DEBUG	WifiScoreReport	1	Score = 60, wifiConnectivityMa
				18:56:30.57	2188	ERROR	adbd	1	failed to connect to socket 'loc
				18:56:30.89	2188	INFO	chatty	1	uid=2000(shell) /system/bin/ac
				18:56:30.99	2188	ERROR	adbd	1	failed to connect to socket 'loc
		AND AT DR.		18:56:31.01	693	ERROR	LOGSERVER_UTILS	1	[Erecovery]readEvent: eRecEver
				18:56:31.10	2188	ERROR	adbd	1	failed to connect to socket 'loc 🔡
				18:56:31.10	17178	WARN	Watchdog	1	hungtask: writing kick
				18:56:31.10	17178	INFO	EventHub	1	EventHub monitor: no key ever 👻
	≡	<u></u>	Ŀ						

### 截屏/录屏

## 打开截屏页签

- 1:在指定设备的远程真机页面,打开截屏页签。
- 2: 在截屏页签, 包括截屏和录屏2个子页签。
- 3:选择**截屏**子页签,查看截屏列表,进行截屏操作;选择**录屏**子页签,查看录屏列表,进行录屏操作。
- 4: 单击大图/小图, 切换截屏/录屏列表预览大小。



#### 截屏

- 1:选择**截屏**子页签。
- 2:在左侧远程真机上操作应用。
- 3: 单击**点击截屏**热区,截取设备模拟器当前界面,并在列表显示。



#### 录屏

- 1:选择**录屏**子页签。
- 2: 单击**开始录屏**热区,开始录屏操作。同时,热区变为**停止录屏**。
- 3:在左侧远程真机上操作应用。
- 4: 单击停止录屏热区,终止录屏操作,并将整个操作的录屏预览在列表中显示。



## 编辑

1: 将鼠标浮动到截屏/录屏预览上方,选中快捷复选框,选中指定截屏/录屏;或单击全选,选中所有截屏/录屏。

2:选中1个或多个截屏/录屏后,单击删除,删除选中的截屏/录屏。



## 上传文件

- 1:在指定设备的**远程真机**页面,打开**文件**页签。
- 2:在**快速进入**文本框,输入完整路径,快速进入指定目录。
- 3: 单击热区选择文件,或将文件拖动至热区,上传文件到指定目录。

# ? 说明

上传的文件须小于256M。



目前暂不支持创建目录,并对目录进行管理操作。

## 设置远程调试

前提条件:已在本地安装adb调试工具。

- 1:在指定设备的远程真机页面,单击更多下拉菜单,选择ADB,打开ADB页签。
- 2:在ADB页签,单击开始adb调试,生成远程真机命令。


3:在本地adb调试工具的命令行执行远程真机命令。

4: 执行 adb devices 命令,查看设备列表,显示当前设备,即可进行远程调试。

```
5: 调试结束后,执行 adb disconnect 命令,断开链接。
```

## 定位测试

- 1. 在指定设备的远程真机页面,单击更多下拉菜单,选择定位,打开定位页签。
- 2. 在左侧远程真机中, 打开已安装应用, 进入应用的待测试页面。
- 3. 设置目标位置。
  - i. 设置经度、纬度,单击确定。

ii. 在位置搜索中输入目标位置,单击Enter。

定位成功后系统提示"虚拟定位成功"。

- ? 说明
  - 远程真机需开启定位权限。
  - 浏览器需开启定位权限,并在浏览器弹窗中单击"允许使用定位选项"。
  - 请确保处于公网环境, 地图需在公网环境下显示。

移动研发平台EMAS / 移动测试 / 远程真机



## 扫码测试

- 1. 在指定设备的远程真机页面,单击更多下拉菜单,选择扫码,打开扫码页签。
- 2. 单击热区选择文件,或将文件拖动至热区,上传文件到指定目录。 上传成功后,系统提示**文件上传成功**。
- 3. 在左侧远程真机中,打开已安装应用,开启二维码扫描功能。

- 4. 扫描方式选择**相册**。
- 5. 在相册中选择待扫描的文件, 系统自动进行扫描。



## 实施性能测试

- 1:在指定设备的远程真机页面,单击更多下拉菜单,选择性能测试,打开性能测试页签。
- 2:在应用下拉列表,选择待测试的应用。
- 3:在进程下拉列表,选择待测试的进程,默认选则主进程。
- 4: 单击⊙,开始测绘指定应用指定进程的性能数据。同时⊙变为●。
- 5: 在左侧设备模拟器操作应用,观察测绘数据的变化。
- 6: 单击 ⓐ, 打开保存到用例右侧栏, 输入任务名称,选择保存的目录和用例, 单击确定, 保存测绘数据至指定目录和用例。

#### 保存到用例 - 任务总能: 2020-11-27 16-38:06 手机率(cont.taebas.taebas) - 用料 ●-- E> WLDTEST ■ WLDTEST ■ WLDTEST ■ WLDTEST ■ ULDTEST新成用制

## ? 说明

目录和用例需预先设置,如未设置,单击**新建用例/新建目录**按钮,即刻添加。新建的目录和用例,后续在**性能测试的用例列表**显示。具体操作请参见: <mark>曾理测试用</mark> 例。

7: 单击 , 完成性能数据的测绘。

## 8: 单击④, 清空已测绘的性能数据。



性能测试报告及相关数据说明,请参见:查看测试报告。

# 6.实施iOS测试

# 6.1. 兼容测试

## 6.1.1. 提交测试

## 前提条件

- 已购买基础测试资源包,或使用量未超过免费阈值。具体说明及操作参见:产品定价。
- (可选)已上传待测试的iOS应用。具体操作参见:上传应用。
- 如应用需登录后启动,需预先设置登录的用例集和参数分组。具体操作参见:新建/导入用例集/新建参数分组。
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面。

## 操作步骤

1:在移动测试控制台的左侧导航栏,选择**测试产品 > 兼容测试**页签。

## 2:在应用下拉列表选择待测试的iOS应用。

#### ? 说明

```
待测试的应用需预先上传;如未上传,单击上传应用,即刻上传。具体操作参见:<mark>上传应用</mark>。
```

## 3: 单击提交测试,打开运行兼容测试页面。

移动研发平台EMAS / 移	动潮试 / 兼容测试				
移动测试	3 4				
测试产品 1 ^	▲ APMOnlineTest ∨      □ 提交測试				
<b>兼容测试</b> 功能测试	概 览 测试报告				
性能测试 New	应用版本: V1.3(1)(2020-02-19 09:57:04) ∨				
远程真机					
测试管理 ~ 解决方案 ~					
计费相关					
帮助					
	<b>暫无測试报告</b> 请点击提交测试马上进行提测				
4. 大学生来的测试					
4: 仕运行兼谷测证	【贝固,按步骤提示,对测试进行配直。具体内容多见:配直说明。 图示20-1-5				
5: 配直元成后,将	鼠你移动 <b>至提父演试</b> ,显示当削测试预计消耗资源已。				
预计消耗资源包 294 台次	· >>购买资源包				
提交测试成功即消耗资源的	9. 无法通回或取消				
上一步提交测试	<b>乱</b> 现2)消				
() NK 85					
1 说明					
<b>预</b> 计消耗贫源包	如趋过当削账尸的免费阈值,或超过已购资源包限额,单击 <b>购买资源包</b> 链接,购买 <b>基础测试</b> 资源包。				
6:单击 <b>提交测试</b> ,	提交测试。				
法择应用及版	*				
	✓ 提交测试成功				
	平台当前还有 32 个任务等待执行,请耐心等待,执行完毕后可重看完整 测试报告。				
	返回兼容性测试				
测试提交后,排队等	穿待执行; 测试执行完毕后,可查看完整测试报告。				
配置说明					



1 BRERUET RELA:	288 ( ) Redott							
▲ 1単 二銀行 ∨ 点開級本:								
V40.5(1)[2521-04-0911+83-00] ···································								
RITHE:								
T-0 Ris								
远坝	10d 09							
应用句	野江洋市已上传/冲扬的ioc应用。可修改							
<b>単用 ビ</b> 私 N 匹 H 口 上 12 / 匹 洋 的 IU 2 座 用 的 『 第 以 。								
	野认诜中已卜传/诜将的应田的版木 如应用在在名个版木 则诜择田干兼空牌测试的版木							
应用版本	② 说明							
	若待测试的应用为混淆版本,请先上传符号表,应用解析成功后再提交测试。上传符号表的具体操作请参见 <mark>上传符号表</mark> 。							
测试备注	输入测试相关信息,便于后续查询回顾。							
测试配直								
→ 选择应用及版本 ————	2 测试配置 (3) 完成							
甘叫信白								
<b>垄</b> 饷 <b>门</b> 忌								
正式版体验版								
服冬雨	於到1面 · ··································							
目定义测试机型 错误日末 自动裁图	◎ 应用安装、后动、卸载 ◎ 投重付货,个限时间及次数 ● 户动时间 CDU上田家 (内方上田家)							
<ul> <li>Generation</li> <li>Genetion</li> <li>Genetion</li> <li>Generation</li></ul>								
□ 选择机型 已选择 10 款								
测试配器								
* 测试时长:								
10	V							
* 登录方式:								
无需登录 用例集登录								
dsvM								
占 近洋工16	4P XIT							
<b>基础信息</b> 分组包括:								
进顶	治明							
起坝	VC 49							
	● 正式 览 · 已购买其砂测试资源句 / 付费使田的版太							
	* <b>工以版</b> ,已对天圣ա网站贝标已,门页区而时版本。							
	⑦ 说明							
使用 <b>正式版</b> 运行兼容性测试,需购买 <b>基础测试</b> 资源包。								
<b>正式版/体验版</b> • 体验版:使用量在免费阈值内,体验使用的版本。								
	⑦ 说明							
	◇ ₩77 <b>存休验版的计费</b> 面列 可含素当前账号的单数额度 · 测试报出的体田를 季控量付费							
	中 <b>行軍隊のアメック</b> 、シューロンスである。 からに、「日本の日本」を行う。							
	购买说明及操作参见: <mark>产品定价。</mark>							

选项	说明
选择机型	<ul> <li>1: 生往选择机型,打开选择提测机型右侧栏, 手动指定或通过预选项批量指定预选机型。</li> <li>IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</li></ul>

## **测试配置**分组包括:

选项	说明
测试时间	如使用 <b>正式版</b> ,则设置测试的时间长度。 ② 说明 如使用体验版,则测试时间默认为10分钟,不可设置。
登录方式	<ul> <li>无需登录:应用无需登录,即可使用。</li> <li>用例集登录:应用需登录后,才可使用。</li> <li>当登录方式设置为用例集登录,需进一步设置,指定登录用例,及相关参数:</li> <li>*周時載巻:</li> <li>0.0.1</li> <li>* 国政時期:</li> <li>● 国际時期:</li> <li>● 国际時期:</li></ul>
用例库版本	<b>当登录方式</b> 设置为用例集登录,指定登录用例所在的用例版本库。 ⑦ 说明 用例库版本需预先设置,或使用默认版本。具体操作参见:新建用例库版本。
测试用例集	当登录方式设置为用例集登录,指定登录用例集所在的用例集。           ⑦ 说明           登录用例集需预先设置;如未预先设置,单击用例管理按钮,打开测试管理>脚本管理页面,即刻进行设置。具体操作参见:新建/导入用例集。

选项	说明			
参数分组	当 <b>登录方式</b> 设置为用例集登录,指定登录用例相关参数分组。 ② 说明 登录用例相关参数分组需预先设置,具体操作参见:新建参数分组。			
账号互踢	当 <b>登录方式</b> 设置为 <b>用例集登录</b> ,设置应用账号是否仅限在唯一终端登录。			
dSYM	压缩ipa包对应的dSYM文件为zip文件,并上传。			
邮件通知地址	设置电子邮箱地址,用于接收测试完成通知。			

### 后续操作

查看概览信息

查看测试报告

## 6.1.2. 查看概览信息

## 前提条件

- 已提交兼容测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

## 操作步骤

- 1: 在移动测试控制台的左侧导航栏,选择测试产品 > 兼容测试页签。
- 2: 在应用下拉列表选择已提交测试的iOS应用,默认打开概览页面。

移动研发平台EMAS / 移动测试 / 兼容测试

移动测	则试	3										
测试产品	1 ^	🗯 UIKitCatalog	∨ ▷ 提5	調试 の上传	应用							
<b>兼容测试</b> 功能测试	2	概览 4式报	概 第 4 武报告									
性能测试	New	应用版本: V13.2(13	3.2)(2019-07-18 16:36:10) 🗸	5						0	•	
远程真机		(00) 常常性测试							+B396 T -	0		
別以官理	č	105兼谷性测试	CTAX						加利すい	2020-12-01 11:44:58	₩ ()	- <u>-</u>
計费相关	·		测试结果错误分布		CRASH		设备数	次数	ANR		设备数	次数
邦助												
ff3 MJ			设备 2 台			暂无数据				暂无数据		
			成功: 2									*

3:在**概览**页面:

- 从应用版本下拉列表中选择应用的版本;
- 从提测于下拉列表中,选择提测时间,默认为最近一次提测;
- 查看指定应用/版本/提测时间的测试概览信息。具体内容参见:概览页面说明
- 4: 单击查看报告按钮, 打开指定测试的详情页面。具体内容参见: 测试报告说明
- 5:如当前测试状态为运行中,单击刷新按钮,手动刷新概览页面,查看最新信息。
- 6: 单击**下载报告**按钮, 下载指定测试的测试报告。

## ⑦ 说明 如当前测试状态为运行中,则下载报告按钮非活性。

## 概览页面说明





## iOS兼容测试概览页面包括:

信息	说明
测试状态	显示当前测试的状态,包括 <b>运行中和已完成</b> 。
测试结果	显示当前测试中运行中/失败/未执行/取消的设备数,及所占比例。
错误分布	显示当前测试中发生的错误类型、影响设备数,及各类错误所占比例。
CRASH	显示当前测试中崩溃异常的统计数据,包括影响设备数和次数。
ANR	显示当前测试中无响应异常的统计数据,包括影响设备数和次数。

## 6.1.3. 查看测试报告

## 前提条件

- 已完成兼容测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

## 查看测试报告

- 1:在移动测试控制台的左侧导航栏,选择测试产品 > 兼容测试页签。
- 2: 在应用下拉列表选择已测试的iOS应用,选择测试报告页签,查看指定应用执行的兼容测试列表。

移动研发平台EMAS / 移动	动测试 / 兼容测试						
移动测试	3						
测试产品 1 ^	🔹 UIKitCatalog	∨ ▷ 提交測试					
兼容測试         2           功能測试	概 览 测试报告	4					
性能测试 (New)	〇 刷新 5						
远程真机 311试等理 5	编号	应用版本	提测时间	測试结果 🕝	状态	▼ 备注	操作
解决方案 🗸	720383	13.2	2020-12-01 11:44:58	2 (2/0/0)	● 已完成	2	查看   分享   删除
计费相关	620410	13.2	2019-07-18 17:04:51	1 (0/ <mark>1/0</mark> )	● 已完成	我是备注我是备注	▲ 查看   分享   删除
帮助							〈 上一页 】 下一页 〉
3:如指定应用有运行中的测试,单击 <b>刷新</b> 按钮,手动刷新 <b>测试报告</b> 页面,查看最新信息。							

4:在测试报告列表的**操作**列:

- 单击查看按钮,查看指定测试的测试报告页面。具体内容参见:测试报告说明
- 单击分享按钮,生成指定测试的测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。
- 单击删除按钮,删除指定测试的测试报告。
- 如测试正在运行中,单击取消按钮,中止测试。

## 6.1.4. 测试报告说明

## 概述

移动研发平台EMAS / 移动测试 / 兼容性测试 / 测试报告详情

← APMOnlineTest(V1.3(1)) ③ 3

 概况
 设备详情
 错误日志
 全部截图
 1



1:选择各页签查看相关内容。

页签	说明
概况	查看当前测试的执行结果、失败类型、失败机型分布、错误日志、失败机型概况。
设备详情	查看当前测试使用的机型,以及测试项目的执行进度和执行结果。
错误日志	查看当前测试的报错信息,包括 <b>崩溃、卡死</b> 和和其他 <b>可忽略异常</b> 。
全部截图	查看当前测试的关键截图。

## 2: 单击右上角各按钮,进行相关操作。

按钮	说明
下载	下载当前测试报告。
分享	生成当前测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。

#### 3: 鼠标移动至帮助按钮, 查看当前测试基本信息。

ID: 720383 測试类型: iOS兼容性測试 提测人: e********* 提测时间: 2020-12-01 11:44:58

## 概况

## 在概况页签,查看当前测试的执行结果、失败类型、失败机型分布、错误日志、失败机型概况。

## 执行结果

## 显示通过/未通过/异常未执行的机型数及其占比。

执行结果							
		类型	占比	机型数量			
_ /	机型总数	通过	60%	6			
	10	未通过	40%	4			
		异常/未执行	0%	0			

## 失败类型

显示安装失败/启动失败/运行失败的机型数及其占比。

失败类型						
	类型	占比	机型数量			
机型总数	安装失败	25%	1			
• 运行失敗: 75%	启动失败	0%	0			
	运行失败	75%	3			

## 失败机型分布

## 按照指定维度,指定状态,显示数量及占比。



选项	说明
维度	按照指定维度,显示各状态的设备数量及占比。 维度包括:品牌/CPU/GPU/ <b>系统版本/内存/分辨率/屏幕</b>
状态	显示指定维度,指定状态的数量及占比。 状态包括 <b>:全部状态/安装失败/运行失败</b>

## 错误日志

## 显示关键错误的列表。

错误日志			
错误类型	出现次数	关键错误	操作
崩溃	8	EXC_CRASH (SIGABRT)	详情
崩溃	1	unknown reason, no call sta	详情
13 查看全部			

单击**查看全部**按钮,或者单击**详情**按钮,打开**错误日志**页签,查看全部错误及其日志详情。具体内容参见:<mark>错误日志</mark>

## 失败机型概况

显示各失败机型的测试结果。

设备	1	安装	7	启动	Ŧ	自动探索	Ŧ	崩溃	卡死	可忽略异常	操作	F
Apple iPhone 6s 11.3.1		•		0		0		1	0	0	详情	1
Apple iPhone XS 12.0		•		0		0	2	1	0	0	<b>3</b> 详惯	ŧ.
Apple iPhone 6 Plus 12.4.5		•		0		0		1	0	0	详情	۲
Apple iPhone 7 Plus 11.0		•		0		0		1	0	0	详情	ŧ.
Apple iPhone 6 Plus 12.0.1		٢		0		0		1	0	0	详情	1
Apple iPhone 6s 12.0		0		0		0		1	0	0	详情	ŧ
Apple iPhone X 11.1.1		٢		0		0		1	0	0	详情	ŧ.
Apple iPhone 5 10.0.1		0		0		0		1	0	0	详情	ŧ.

1:在列表的表头,单击**安装/启动/自动探索**列的筛选按钮,对列表内容进行筛选。

2: 在崩溃/卡死列,显示设备的异常个数,单击数字链接,查看异常的详细信息。

3:在操作列,单击详情按钮,打开设备详情页面。具体内容参见:设备详情

## 设备详情

在**设备详情**页签:

1:在左侧**设备**列表,单击筛选下拉按钮,按照**分类方式**或**状态**筛选设备。

筛选项	说明
分类方式	将测试设备按照指定分类方式进行分组,包括 <b>按测试结果、按系统版本、按品牌</b> 3个选项。
状态	将测试设备按照指定状态进行筛选,包括 <b>全部、未通过、设备错误、异常/未执行、超时、通过、等待中、运行中</b> 。 当 <b>状态</b> 设置为 <b>未通过</b> 选项,可进一步按照 <b>失败环节</b> 进行筛选,包括 <b>安装、启动、运行、卸载</b> 4个环节。

2: 在左侧设备列表,单击设备的名称,在右侧打开指定设备的详细信息。

概况 设备详情 1	日志	全部截图							
设备 2 7	7 筛选	Apple - iPhone 6s	Plus - 11.4	通过					
- L		▲ 曲株	🔮 启动		🕝 Ripper智能	探索		2 卸载	
∨ 未通过	分类方式:								
<ul> <li>Apple - iPhone 6s - 11.3.</li> </ul>	按测试结	果(默认) 🗸	斤						
<ul> <li>Apple - iPhone XS - 12.0</li> </ul>									
Apple - iPhone 6 Plus - 11	状态:								
Apple iPhone 7 Plus 11	全部	$\sim$	@ 1% 🗆 f	£88 ♥     	¥46.62	₽ 1% <b></b> +	无服务 🅈	F#5-51 EMAS	@ 1% 🕞 f
<ul> <li>Apple - IPhone 7 Plus - 1</li> </ul>				补丁测试			[重要]应用配置信息	地震	
<ul> <li>Apple - iPhone 6 Plus - 12.0</li> </ul>	0.1	Abort		补丁加载测试			API周关		
Apple - iPhone 6s - 12.0		上报JSError		PATCHIER			Weex		
Apple - iPhone X - 11.1.1		主动上报tlog		Patch拉取			高可用		
Apple - iPhone 5 - 10.0.1		检查远程日志任务		Patch清空			热修复		
<ul> <li>Apple - IF Ione 5 - 10.0.1</li> </ul>		触发卡顿		PATCHIBILI			送程配置		
∨ 通过	1	上报Full Trace	Notice and Table	本地补丁载入			ACCS		
			47791051104 P18	二维码扫描					
Apple - iPhone 6s Plus - 1	1.4	Lincent							
							Favoritas	C	et e
		state0			state2			state1	

## 执行状态

显示指定设备各兼容性测试项(安装/启动/智能探索/卸载)的执行状态。

Apple - iPhone 6 -	12.4.8 通过			
⊘ 安装	❷ 启动	✓ Ripper智能探索	❷ 卸载	

如指定设备的测试未通过,单击**远程真机**按钮,申请使用相同机型设备,进行测试结果验证。远程真机具体操作参见:远程调试

## 智能探索

显示测试过程的关键截图/视频,和异常汇总。

看截图						口 视频	设备
5:54 1 🕈 🚥	5:53	🕈 👪	5:51 1	🗢 👀	5:52	🕈 🐝 5:52	
EMAS	く远程配置		Keex Demo Weex Demo		EMAS	<pre>&lt; EMAS</pre>	
全堆栈Crash	配置列表	刷新			[重要]应用配置信息检查	补丁测试	
Abort					API网关	补丁加载测试	
上报JSError					Weex		
主动上报tlog					高可用	PATCH加载	
检查远程日志任务					热修复	Patch拉取	
鼬发卡顿					远程配置	Patch清空	
上报Full Trace					4000	PATCH调试工具	1
					ACCS	本地补丁载入	
应且循环分用(切卢古土液,…						-44-171+71+#	
上报埋点							
					Pavorites Q Search		
stateO	state6		state5		state8	state7	

在**查看截图**区域:

- 单击视频/截图切换按钮,查看智能探索过程的视频/截图。
- 单击设备日志按钮,查看/下载指定设备的日志详情。

异常汇总				
序号	错误类型 🕎	名称	总次数 🖕	操作
1	崩溃	unknown reason, no call stack available!	1	详情 ∨

王 <b>异常汇总</b> 列表,单击 <mark>操作</mark> 列 <b>生能分析</b> 查看应用运行过程中,CPU/内·	的 <b>详情</b> 按钮,展开指定错误的详细信息。 存/网络资源占用、运行情况的统计数据和B	<b>寸间趋势,以及细化至进程粒度的资源</b>	占用情况。	
智能探索 性能分析				
性能概况 ^{CPU} 占用率(%) <b>2</b> .86	内存占用(MB) <b>374</b> .25	网络耗用(KB/min) <b>72</b> .59	丢帧率(%) <b>13</b> .89	
		11:39:26		
150 75 0 1800 600 0 11-25 11:35:43	CPU     内容     网络     无限     电显     11-25 11:37:43     11-25 11:39:43	J: 0 % F: 379.79 MB   PSS: 379.79 MB   HeapA 8: 24,931.06 KB   上行: 735.63 KB 下行 章率: 0 % 1: 692.79 mA   CPU: 422.79 mA   WiFi: 11-25 11:41:43	lloc: 165.07 MB f: 24,195.44 KB 110 mA 屏幕: 160 mA 11-25 11:43:43	 11-25 11:45:43
11-25 11:35:43 (鼠标移动至时间趋势图上方, <b>彗误日志</b> (错误日志页签:	- CPU占用率(%) - 内:	存(MB) — 网络(KB) — 丢帧率(%)	— 电量(mA)	<u>11-25 11:46:42</u>
· 投訊崩(以下死) 可恐喝并( : 错误日志默认显示部分内容 : 单击下载错误日志按钮,  : 单击关联设备按钮, 查看引 概况 设备详情 错误日志	● 57 英, 量有有读口心。 8、 单击扩展按钮,查看完整日志。 下载指定的错误日志。 引发当前错误的设备;从下拉列表中单击设备 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	备名称,打开指定设备的 <b>设备详情</b> 页面	ī.	
全部典型 崩溃(2) 崩溃(共8次): EXC_CRASH(1) Last Exception Backtrace: のに4872fcff8 のに4872fcff8 のに4872faf4 のに487df44 のに487df4bc のに487df4bc のに487df7bc	- 可忽略异常 (0) 2 SIGABRT)			▲ 公規書 法目表 法规定 >
崩溃 (共1次): unknown reasc {"app_name":"EMASDemo","timesta DE62449E4344","adam_id":@"buil (IGA366)","incident_id":*88C4A Date/Time: 2020-11-13 17: End time: 2020-11-13 17: GS Version: iPhone GS 12.0 Architecture: arm64e	pn, no call stack available! mp":"2020-11-13 17:56:34.65 +0800","app_version": d_version":"1.0.0","bundleID":"com.emas.poc12xjc" &G-988-4F1-A302-698A37038842","name":"EMASDemo"] 56:23 +0800 (Build 16A366)	'1.0.0.10631","slice_uuid":"0E9F0278-F76C-34 "share_with_app_devs":true,"is_first_party"	38-8947- :false,"bug_type":"202","os_vers	下载错误日志: 关联设备 > tion":"iPhone OS 12.0

全部截图

概况	设备详情 错误日志	全部截图 1		
日前	选 2			
分类	<b>方式:</b> 辨率 ── ── ──	3		
15	36x2048 V		B.A.B.W	

#### 在**全部截图**页签:

- 1:单击筛选按钮,设置分类方式及其取值,查看指定分类的截图。
- 2: 单击右侧大图/小图切换按钮,按照指定方式显示截图。

## 6.2. 功能测试

## 6.2.1. 提交测试

## 前提条件

- 已购买基础测试资源包,或使用量未超过免费阈值。具体说明及操作参见:产品定价。
- 已上传待测试的iOS应用。具体操作参见:上传应用。
- 已创建/导入功能测试的用例集。具体操作参见:新建/导入用例集。
- 已上传/在线录制脚本,并创建用例。具体操作参见:新建用例。
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面。

#### 操作步骤

- 1:在移动测试控制台的左侧导航栏,选择测试产品 > 功能测试页签。
- 2:在应用下拉列表选择待测试的iOS应用。

## ⑦ **说明** 待测试的Android应用需预先上传;如未上传,单击**上传应用**按钮,即刻上传。具体操作参见:上传应用。

#### 3: 单击提交测试按钮, 打开运行功能测试页面。

移动研发平台EMAS / 移动	动测试 / 功能测试
移动测试	8 4
测试产品 1 ^	▲ APMOnlineTest ∨ 区 提交測试 の 上传应用
兼容测试 功能测试 2	概 览 测试报告
性能测试 Now	应用版本: V1.3(1)(2020-02-19 09:57:04) ∨
远程具机 测试管理 ~	
解决方案 💙	
> 计费相关	
YN BU	<b>暂无测试报告</b> 请点击跟交测试马上进行强测
4: 在运行功能测试 5: 配置空成后 悠	贞面,按步骤提示,对测试进行配置。具体内容参见: <mark>配置说明</mark> 。 鼠标路动 <b>至提                                    </b>
5. 癿直元成归,将	
预计消耗资源包 294 台次 提交测试成功即消耗资源包	>>>购买资源包 ), 元法退回或取消
上一步提交测试	TROM
② 说明	

预计消耗资源包如超过当前账户的免费阈值,或超过已购资源包限额,单击**购买资源包**链接,购买基础测试资源包。

6: 单击**提交测试**按钮, 提交测试。

→ 选择应用及版本	→ 刻试配置	- 3 完成
	✔ 提交测试成功 平台当前还有 5 个任务等待执行,请耐心等待,执行完毕后可重看完整测	
	试报告。 返回功能测试	

## 测试提交后,排队等待执行;测试执行完毕后,可查看完整测试报告。

## 配置说明

## 选择应用及版本

编位用及版本	(z) medem	
80000000000000000000000000000000000000		
15		

# 选项 说明 应用包 默认选中已上传/选择的应用,可修改。 应用版本 默认选中已上传/选择的应用的版本。如应用存在多个版本,则选择用于功能测试的版本。 ② 说明 若待测试的应用为混消版本,请先上传符号表,应用解析成功后再提交测试。上传符号表的具体操作请参见上传符号表。 测试备注 输入测试相关信息,便于后续查询回顾。

## 测试配置

基础信息		
正式版 体验版		
服务项	检测项	计费项
✓ 自定义测试机型	📀 应用安装、启动、用例执行、卸载	按量付费,不限时间及次数
⊘ 用例测试结果	⊘ 启动时间、CPU占用率、内存占用率	(本月剩余体验 19981 台次)
✓ Crash/ANR检测	❷ 视频、截图	
✓ 账号池		

## □ 选择机型 已选择 10 款

## **基础信息**分组包括:

选项	说明				
正式版/体验版	<ul> <li>正式版:已购买基础测试资源包,付费使用的版本。</li> <li>② 说明 使用正式版运行功能测试,需购买基础测试资源包。</li> <li>体验版:使用量在免费阈值内,体验使用的版本。</li> <li>③ 说明 在体验版的计费项列,可查看当前账号的免费额度;测试超出的使用量,需按量付费。</li> </ul>				
	购买说明及操作参见: <mark>产品定价</mark> 。				

选项	说明
选择机型	<ul> <li>1. 生 生 选择 机 型 控 包 、 打 开 选 择 提 测 机 型 石 侧 栏 、 手 动 指 定 或 通 过 预 读 远 过 过 过 过 过 过 过 过 过 过 过 过 过 过 过 过 过 过</li></ul>

## 测试配置

▶ 用例库版本:			
1.0.0			V
测试用例集:			
选择用例集	用例管理		
用例集名称	描述	所属目录	操作
WLDTEST_iO	S测试	WLDTEST- iOS测试模 块	移除
参数分组:			
默认参数空间			~
长号互踢:			
否			
ISYM :			
上 选择上传	提示: 仅支持选择.z	rip 文件	
青将ipa包对应的 邮件通知地址:	 dSYM文件 压缩成 zip文t	件 进行上传	
邮件通知地址			

## **测试配置**分组包括:

选项	说明
用例库版本	指定功能测试用例集所在的用例版本库。
测试用例集	指定功能测试用例集。 ⑦ 说明 功能测试用例集需预先设置;如未预先设置,单击用例管理按钮,打开测试管理 > 脚本管理页面,即刻进行设置。具体操作参见:新建/导入用 例集。
参数分组	指定功能测试用例集相关的参数分组。 ⑦ 说明 登录用例集相关参数分组需预先设置,具体操作参见:新建参数分组。

选项	说明		
账号互踢	设置应用账号是否仅限在唯一终端登录。 ⑦ 说明 账号互踢设置为是时,需在用例管理页面的参数池页签中配置参数account,以添加相关账号。配置参数的具体操作请参见 <mark>新建/导入参数</mark> 。		
dSYM	压缩ipa包对应的dSYM文件为zip文件,并上传。		
邮件通知地址	设置电子邮箱地址,用于接收测试完成通知。		

## 后续操作

查看概览信息

查看测试报告

## 6.2.2. 查看概览信息

## 前提条件

- 已提交功能测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

#### 操作步骤

- 1:在移动测试控制台的左侧导航栏,选择**测试产品 > 功能测试**页签。
- 2:在应用下拉列表选择已提交测试的Android应用,默认打开概览页面。

移动研发平台EMAS / 移动测试 / 功能测试

移动测试	3					
测试产品 1 ^ 兼容测试 功能测试 2	<ul> <li>▲ APMOnlineTest</li> <li>● 上传函</li> <li>概览</li> <li>④ 批报告</li> </ul>	7月				
性能測试 <b>New</b> 远程真机 測试管理 <b>&gt;</b>	应用版本: V1.3(1)(2020-02-19 09:57:04)∨ 5 ioS功能测试 (運行中)				6 提测于: 2020-12-02 11:19:56	7 × 111 O L
解决方案 🖌 🖌	用例结果 错误分布		用例集	模块		成功 / 失败 / 异常
) 计费相关		-	WLDTEST_iOS测试	WLDTES	-iOS测试模块	3 / 3 / 4
常助			设备名称		測试结果	
	成功: 3		Apple iPhone 6 12.4.8		通过	
			Apple iPhone SE 12.2		未通过	
			Apple iPad8,1 12.1.4		未通过	<b>P</b>
			Apple iPhone 5s 12.4		通过	8
	失败: 3		Apple iPad air2 14.2		通过	

3:在**概览**页面:

从应用版本下拉列表中选择应用的版本;

从提测于下拉列表中,选择提测时间,默认为最近一次提测;

查看指定应用/版本/提测时间的测试概览信息。具体内容参见:概览页面说明

4: 单击查看报告按钮, 打开指定测试的详情页面。具体内容参见: 测试报告说明

5:如当前测试状态为运行中,单击刷新按钮,手动刷新概览页面,查看最新信息。

```
6: 单击下载报告按钮, 下载指定测试的测试报告。
```

## ⑦ 说明 如当前测试状态为运行中,则下载报告按钮非活性。

## 概览页面说明



#### iOS功能测试概览页面包括:

信息	说明
测试状态	显示当前测试的状态,包括运 <b>行中</b> 和 <b>已完成</b> 。
用例结果	显示当前用例中运行中/失败/未执行/取消的设备数,及所占比例。
测试结果	基于每测试设备,显示当前测试的测试结果。

## 6.2.3. 查看测试报告

## 前提条件

- 已完成功能测试。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见: 打开控制台页面

## 查看测试报告

- 1:在移动测试控制台的左侧导航栏,选择测试产品 > 功能测试页签。
- 2:在应用下拉列表选择已测试的Android应用,选择测试报告页签,查看指定应用执行的功能测试列表。

移动研发平台EMAS / 移	动测试 / 功能测试						
移动测试	3						
测试产品 1 ^	KAPMOnlineTest	∨ 区 提交測	试 ↔ 上传应用				
兼容测试 功能测试 2	概 览 测试报	告 4					
性能测试 (New)	〇 刷新 5						
远程真机	编号	应用版本	提测时间	測试结果 🕝	状态	▼ 备注	操作
刻山官理 V 解决方案 V	720943	1.3	2020-12-02 11:19:56	10 (3 / 4 / 3)	● 运行中	2	6 查看   分享   取消
> 计费相关 帮助							〈上一页 】 下一页 〉
3: 如指定应用有运	行中的测试,单	击 <b>刷新</b> 按钮,手动刷新	<b>测试报告</b> 页面,查看最新	新信息。			

4: 在测试报告列表的操作列:

- 单击查看按钮,查看指定测试的测试报告页面。具体内容参见:测试报告说明
- 单击分享按钮,生成指定测试的测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。
- 如测试正在运行中,单击取消按钮,中止测试。

## 6.2.4. 测试报告说明

概述	
移动研发平台EMAS / 移动测试 / 现试报告详情	
← APMOnlineTest(V1.3(1)) <a>Image: 3</a>	(已完成) 下载   分享   2
概况 设备详情 错误日志 全部截图 1	
1:选择各页签查看相关内容。	

页签	说明			
概况	查看当前测试的执行结果、错误日志、用例集概况。			
设备详情	基于每测试设备,查看/编辑测试用例的执行结果,及相关日志。			
错误日志	查看当前测试的报错信息,包括 <b>崩溃、卡死</b> 和和其他 <b>可忽略异常</b> 。			
全部截图	查看当前测试的关键截图。			

#### 2: 单击右上角各按钮,进行相关操作。

按钮	说明
下载	下载当前测试报告。
分享	生成当前测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。

## 3: 鼠标移动至帮助按钮, 查看当前测试基本信息。

ID: 715517 測试类型: iOS功能測试 提测人: m********* 提测时间: 2020-11-13 19:50:43 測试账号互踢: 否

## 概况

在概况页签,查看当前测试的执行结果、错误日志、用例集概况。 执行结果 显示通过/未通过/异常未执行的机型数及其占比。

执行结果			
	类型	占比	机型数量
	19114		

机型总数 10	通过	30%	3
10	未通过	40%	4
	异常/未执行	30%	3

## 错误日志

显示关键错误的列表。

错误日志			
错误类型	出現次数	关键错误	操作
崩溃	7	EXC_CRASH (SIGABRT)	详情
□ 查看全部			

单击**查看全部**按钮,或者单击**详情**按钮,打开错误日志页签,查看全部错误及其日志详情。具体内容参见:错误日志

用例集概况

基于每用例集,显示当前应用的测试结果。

用例集概况					
用例集名称	用例库版本	目录	用例集类型	成功/失败/异常(机型)	<b>*1</b> 2
WLD用例集	1.0.0	WLDTEST	录制用例	3 / 5 / 2	用例概况 🗸 机型概况 🗸

1:在用例集列表,单击操作列的用例概况按钮,显示用例集用例的详细信息。

用例集機	既况							
	用例集名称	用例库版本	目录	用例集类型	成功/	/失败/异常(机型)	操作	
	WLD用例集	1.0.0	WLDTEST	录制用例	3 / 5	5/2	用例概况 ヘ   机型概况 >	
	用例名称		用例描述			脚本名称	操作	
	WLDTEST在线录制用例		用户指南文档开发测试用例			WLDTEST在线录制脚本	查看	

## 2: 单击**机型概况**按钮,基于每测试设备,显示测试结果。

用例集構	既况								
	用例集名称	用例库版本	目录	用例集类型	成功/失败/异常	(机型)	操	作	
	Demo用例集	0.0.1	默认模块	录制用例	6 / 4 / 0			用例概况 ∨   机型概况	^
	设备名称			状态		安装	执行	操作	
	Apple iPhone 6s 12.0			通过		0	0	执行日志   设备详情	
	Apple iPhone X 11.1.1			未通过		٥	0	执行日志 🗌 设备详情	
	Apple iPhone 7 Plus 11.4.1			通过		0	0	执行日志   设备详情	
	Apple iPhone XR 12.0.1			通过		٥	0	执行日志   设备详情	
	Apple iPhone 6 12.4.5			未通过		0	0	执行日志   设备详情	
	Apple iPhone 5s 11.4			通过		٥	0	执行日志   设备详情	
	Apple iPhone 8 11.1			通过		0	ø	执行日志   设备详情	
	Apple iPhone11 13.0			通过		•	0	执行日志   设备详情	
	Apple iPhone 7 10.3.2			未通过		0	0	执行日志   设备详情	
	Apple iPhone 8 Plus 13.5.1			未通过		٥	0	执行日志   设备详情	

3:在**机型概况**列表,单击**执行日志**按钮,查看指定设备的执行日志;单击**设备详情**按钮,打开指定设备的**设备详情**页面,查看详细信息。具体内容参见:设<mark>备详</mark>情

## 设备详情

### 在**设备详情**页签:

1:在左侧**设备**列表,单击**筛选**下拉按钮,按照**分类方式**或**状态**筛选设备。

筛选项	说明
分类方式	将测试设备按照指定分类方式进行分组,包括 <b>按测试结果、按系统版本、按品牌</b> 3个选项。
状态	将测试设备按照指定状态进行筛选,包括 <b>全部、未通过、设备错误、异常/未执行、超时、通过、等待中、运行中</b> 。 当 <b>状态</b> 设置为 <b>未通过</b> 选项,可进一步按照 <b>失败环节</b> 进行筛选,包括 <b>安装、启动、运行、卸载</b> 4个环节。

## 2: 在左侧设备列表,单击设备的名称,在右侧打开指定设备的详细信息。

	概况 设备详情 1	日志全部截图		
	设备 重跑	Apple - iPhone 5s 又 筛选 2	- 12.4 (通过)	♪ 远程真机
		- ets 14+	Ø WLDTEST_iOS测试	
	◇ 木通辺	分类方式:	→ WLDTEST iOS测试 ····	
	<ul> <li>Apple - iPhone se2 - 14.2</li> </ul>	按测试结果(默认) 🗸		
	<ul> <li>Apple - iPhone 6 - 11.4.1</li> </ul>	状态:		
>	<ul> <li>Apple - iPad8,1 - 12.1.4</li> </ul>	全部 、	测试 😢 00:00:24 🕞 視频	▶ 脚本日志   ▶ Appium日志   ▶ 设备日志
	<ul> <li>Apple - iPhone SE - 12.2</li> </ul>		制脚本 😆	
	∨ 通过	3		
	Apple - iPhone 6 - 12.4.8	县堂汇总		
	Apple - iPad air2 - 14.2	_		
	• Apple - iPhone 5s - 12.4	<u></u> ^{序号}	错误类型 👕 名称	总次数 🔶 操作
	◇ 运行中	1		
	<ul> <li>Apple - iPhone 6 Plus - 12.</li> </ul>	4.5		
	∨ 异常/未执行	2	暂无数据	
	Apple - iPhone 6 - 10.2			

## 执行状态

显示指定设备各用例集的执行状态。

Apple - iPhone X - 11.1.1 未通过

如指定设备的测试未通过,单击远程真机按钮,申请使用相同机型设备,进行测试结果验证。

用例集执行概况

1:在查看用例集执行结果下拉列表,选择用例集,查看指定用例集的执行概况。

2:单击视频按钮,查看指定用例集的执行视频记录;单击脚本/Appium/设备日志按钮,查看/下载指定用例集的各类执行日志。

查看用例集执行结果:	$\sim$	1

用例集执行概况	
區 WLDTEST_iOS测试 ❷ 00:00:24	2 🖸 视频 🕒 脚本日志 🕒 Appium日志 🖙 设备日志
> WLDTEST-iOS录制脚本 ❷	

#### 异常汇总

查看指定用例集的异常信息。

异常汇总				
序号	错误类型 🛒	<b>1</b>	总次数 🍦	操作
1	崩溃	EXC_CRASH (SIGABRT)	1 2	详情 ∨

#### 在**异常汇总**列表:

- 单击错误类型列的筛选按钮,按照指定错误类型,筛选异常。
- 单击操作列的详情按钮,展开指定错误的详细信息。

#### 性能分析曲线

查看指定用例集在运行过程中,CPU占用率/内存/网络/FPS随时间变化的趋势。





## 错误日志

在**错误日志**页签:

- 1:按照**崩溃/卡死/可忽略异常**的分类,查看错误日志。
- 2:错误日志默认显示部分内容,单击扩展按钮,查看完整日志。
- 3: 单击下载错误日志按钮,下载指定的错误日志。
- 4: 单击管理设备按钮, 查看引发当前错误的设备。

概 况 设备详	¥情 错误日志	1 『截图				
全部类型	崩溃(1) 卡	死 (0)	可忽略异常 (0)			3 4
崩溃 (共 7 次) Last Exception 0x1e3e17778 0x1e3010284 0x1e3d2b5e0 0x1e3e1d9e4 0x1e3e1f750 0x1028aa8d4	: EXC_CRASH (SI Backtrace:	GABRT)				下載错误日志 关联设备 ∨

## 全部截图

マ 筛选 2									3 💷 🗄
分类方式:	-	-**	7.814	281 _+100 <(1045 直用配置信息	241+∞ <2045	741+100 <€5555 £54₽₽##	2524++ 100 EMAS	7.01	241♥ 00 < 85445
			[重要]应用配置信息检查	LAsskey/Fill Appliev 10000016 LAssEncer/Fill Applievel	全堆栈Crash	Deploy VIII Applies 100000016 Deploy w/VIII Applies et	[重要] 应用配置信息检查	10000016 10000016 Destaure TELAstance	应用配置值 全堆栈Crash
品牌	$\sim$		API周关	es: #5c8093dabe875a04e3276203e2 (Accsower#81ece88 enseptio-acs.emas-ha.cn	Abort	escrectore PE exet a contract of the contract	API 民关	ddc115c8b69Aa6e875a04e3278203le2 Sectoreminitia ecette emaspociacsiemasiha.cn	¥ Abort
		20 X00 X8	Weex	(International Web model) emanpoid-asserver.emas-ha.cn (Developed)#80	上用JSEnor	(http://www.initial.com/ emanpoc-assrver.emas-ha.cn (http://www.initial.com/	Weex	Microsometrik, molek emaspoc-aserver.emas-ha.cn (Genetikitik) i i iko	上报JSError
Apple	$\sim$		高可 允许"EMAS4POC"使	1001(#POC_JOS_1.0.0 (20x1w/M_TH) Here MSBBB Mtp://publish-poc.emas-ha.cn/eveex/	主动上报tlog	tites ft许"EMAS4POC"使	高可用	NON 光祥"EMAS4POC"使	) 主政①报40g
		feetberk methods, arguiterer (histolik)	用无线数据? 共成无线数据时,部分改成	enas-ha-remote-log-poc	检查这程日志任务	14000 用大线数据? 1973年 未过大线数据时,部分功能	热修复	enas 用无线数据? enas ACCTALEND, IFSIDE	(检查送程日志任务
	12112.00.00		SE SECTION.	[INService(PEE] IA Service (0 [INSErviceanter(FEE] etech(E)]	触发卡顿	Delay URCERCH.	送程配置	passa naccitata	二 触发す頓
	ACCS		ACC CREEMIN	Introduction TECHNEREE STRATCHEE emission-aperveriences he co	上报Full Trace	enter GREMEN	ACCS	OWNER CREASE	上报Full Trace
			不允许	新天役置 1955-50-10(中北)中北(市市安地北)	检查循环引用(切断台上接,	<b>第天日</b> 不允许		WFEI 不允许	检查循环引用(切后
				Interviewel as service and an interviewel	上报理点	Interview association and an and a second as a second		Severative and the second seco	上报埋点
				WerScuthkyEDAM(weiMySchill		WHITELENDING COMMON MATCH		W+rSzs@H4yEDAMje+1MiSzAF	
				1.0.0.10631 (#1044;410)#88 10.5:12 III		1.0.0.10631 (#1044g/192) #888 10 % 02 88		10.0.10631 (mmmacFill) #第8 報志会習	
	* 9	🔍 🕗 🖸 🞵	* <u>Q</u>				* Q		*
							Paradhan Starth		
ple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR
ple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR
ole iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR
Line iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR 78 78 78 78 78 78 78 78 78 78 78 78 78	Apple iPhone XR	Apple iPhone XR
Le iPhone XR	Apple iPhone XR Zaz 全球低Cassh Abort	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR  781  781  781  781  780  780  780  780	Apple iPhone XR	Apple iPhone XR
CONTRACTOR      CONTRACTO	Apple iPhone XR 232 全球机会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会会	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR 2010 2010 2010 2010 2010 2010 2010 2	Apple iPhone XR	Apple iPhone XR 732 -*** Cturas 2318/Crash Abort 1:18/J5fmrr	Apple iPhone XR  72  72  72  72  72  72  72  72  72	Apple iPhone XR	Apple iPhone XR Apple iPhone XR Bus app
the information of the info	Apple iPhone XR 782 Closs 28ECanh Abot 18.5Enor 180.5Enor	Apple iPhone XR The state of the second s	Apple iPhone XR	Apple iPhone XR  723 -*** Cluss  228Cresh Abor LBJSfrere 3282Lfilesg	Apple iPhone XR  751  153  154  154  154  154  154  154  1	Apple iPhone XR  283 Citats	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR Apple iPhone XR Internet of the second seco
Comparison of the second	Apple iPhone XR ⁷²⁸ - **** 全然化Cean Appt 上形 JSforer 主形 JSforer 主形 JSforer 主形 JSforer 主称 JSforer	Apple iPhone XR	Apple iPhone XR	Apple iPhone XR  Apple iPhone XR  Settors  Class Settors  Acor  LSuScen Settors  thttp://glib.fr/s	Apple iPhone XR	Apple iPhone XR	AppleiPhone XR	Apple iPhone XR	Apple iPhone XR
	Apple iPhone XR tas Class Abot LB.Rfmor Stat LBroge the able of the filled Rec.+M	Apple iPhone XR 721	Apple iPhone XR	Apple iPhone XR  201 202 203 203 203 203 203 203 203 203 203	Apple iPhone XR	Apple iPhone XR	Antonio de la construir de la	Apple iPhone XR	Apple iPhone XR
	Apple iPhone XR  ***  ***  ***  ***  ***  ***  ***	Apple (Phone XR	Apple iPhone XR	Apple IPhone XR 2400 24150wh 25170wh 25170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 15170wh 1517	Apple Phone XR	Apple iPhone XR 30 30 -*** (una 20 20 20 20 20 20 20 20 20 20 20 20 20	Allow we have a set of the set of	Apple iPhone XR	Apple iPhone XR
Lector Control Control Contro	Apple iPhone XR	Apple IPhone XR	Apple IPhone XR	Apple IPhone XR	Apple IPhone XR	Apple iPhone XR	A A A A A A A A A A A A A A A A A A A	Apple iPhone XR	Apple iPhone XR  State of the second seco
A second	Apple iPhone XR  ***  *** *** *** *** *** *** *** *	Apple iPhone XR Total State State State State State State State State State State State State State State State State State State State State State State State State State State State State State	Apple iPhone XR	Apple IPhone XR  **  **  **  **  **  **  **  **  **	Apple IPhone XR	Apple iPhone XR  xa -*** (was 2480cm 2480	And the second s	Apple iPhone XR	Apple iPhone XR
A contract of the second secon	Apple iPhone XR  *** *** *** *** *** *** *** *** **	Apple IPhone XR	Apple IPhone XR	Apple iPhone XR *** *** *** *** *** *** *** *** **		Apple iPhone XR *** *** *** *** *** *** *** *** **		Apple iPhone XR	Apple iPhone XR
Le inhone XR	Apple iPhone XR	Apple IPhone XR	Apple iPhone XR	Apple iPhone XR	Apple IPhone XR	Apple IPhone XR  ***  *** *** *** *** *** *** *** *		Apple iPhone XR	Apple iPhone XR

在**全部截图**页签:

1:单击筛选按钮,设置分类方式及其取值,查看指定分类的截图。分类包括品牌/Activity/分辨率/系统版本4个选项。

2: 单击右侧大图/小图切换按钮,按照指定方式显示截图。

## 6.3. 性能测试

## 6.3.1. 管理测试用例

本文介绍管理性能测试用例的操作方法。

## 背景说明

执行性能测试时,在开始测试前或者测试结束后需要新建用例,用于保存测试数据。您可以申请远程真机进行性能测试,测试过程请参见<mark>实施性能测试</mark>。

## 前提条件

已打开移动测试控制台页面。具体操作参见:<mark>打开控制台页面</mark>。

## 打开性能测试页面

## 1:在移动测试控制台,选择**测试产品 > 性能测试**,打开**性能测试**页面。

2:在应用下拉列表,选择待测试的iOS应用。

```
移动研发平台EMAS / 移动测试 / 性能测试
移动测试
             3
测试产品 1 ^
            🗯 APMOnlineTest
                            \sim
 兼容测试
              用例列表
 功能测试
性能测试 🔤 🙎
 远程真机
                                                                   选择用例查看
测试管理
                                                                您可选择用例查看或者直接运行测试
                   暂无用例数据
解决方案
         ~
                                                                     运行测试
                    请新建用例
计费相关
                 新建用例 新建目录
帮助
```

## 新建目录

1. 在**性能测试**页面,如**用例列表**为空,即初次创建目录,单击**新建目录**按钮;或从**用例列表**的扩展菜单中,选择**新建目录**选项,打开**新建目录**对话框。



2. 在**新建目录**对话框,填写目录信息。

属性	说明
父级目录	用于指定目录的存储路径,即所在父级目录。
目录名称	用于设置目录的名称。 【取值范围】2~50字符 【字符类型】中文/英文大小写/数字/下划线/短横线
	同级目录的名称不能相同。

## 3. 单击**确定**,保存目录。

目录创建后,返回**性能测试**页面,在**用例列表**查看已创建的目录。

APMOnlineTest	V	🛛 运行测试
用例列表		
▶ □ WLDTEST-IOS目录		

## 新建用例

- 1. 新建用例包含两个场景:
  - 已新建目录:将鼠标滑动至用例列表中已创建的目录上,从快捷菜单中选择新建用例选项,在指定目录下创建用例。



○ 未新建目录:在性能测试页面,如用例列表为空,即初次创建用例,单击新建用例按钮;或从用例列表的扩展菜单中,选择新建用例选项,打开新建用例对活框。



	属性	说明
--	----	----

属性	说明
父级目录	用于指定用例的存储路径,即所在父级目录。 ⑦ 说明 新建用例后,可使用鼠标左键拖动用例,更改父级目录。
用例名称	用于设置用例的名称。 【取值范围】2~50字符 【字符类型】中文/英文大小写/数字/下划线/短横线

3. 单击**确定**,保存用例信息。

## 管理用例列表

1. 将鼠标移动到已创建的目录上,单击快捷按钮,对目录进行管理。

用例列表	:
▼ 🗁 zz	⊕ <u>∠</u> ×
🗋 cc	新建用例
	新建目录

选项	子项	说明
	新建用例	在当前目录下,新建用例。
	新建目录	在当前目录下,新建子目录。
2	-	重命名当前目录。
		删除当前目录。
×	-	注意 删除目录后,目录下所有用例将被同时删除,且数据不可恢复,请谨慎操作。

2. 将鼠标移动到已创建的用例上,单击快捷按钮,对用例进行管理。

用例列表 ▼ B WLDTEST目录 C WLDTEST用例	
选项	说明
2	重命名当前用例。
	删除当前用例。
x	注意     删除用例后,数据将不可恢复,请谨慎操作。

3. 选中已创建的目录/用例,长按鼠标左键,并将目录/用例拖动到适当位置。

## 后续操作

## 提交测试

查看测试报告

## 6.3.2. 提交测试

## 前提条件

- 已购买远程真机资源包,或使用量未超过免费阈值。具体说明及操作参见:产品定价。
- (可选)已上传待测试的iOS应用。具体操作参见:上传应用。
- (可选)已创建测试目录和用例。具体操作参见:管理测试用例。
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面。

#### 操作步骤

## 1:在移动测试控制台的左侧导航栏,选择测试产品>性能测试页签。

2:在应用下拉列表选择待测试应用。

? 说明

待测试的应用需预先上传,具体操作参见:上传应用;如未上传,待后续操作过程中上传。

## 3:在用例列表选择已创建的用例。

## 🥐 说明

```
测试目录和用例需预先设置,具体操作参见:管理测试用例;如未设置,待后续操作过程中设置。
```

## 4: 单击运行测试按钮, 打开远程真机页面。

移动研发平台EMAS / 移动	测试 / 性能测试						
移动测试	3						
测试产品 1 ^	🔹 APMOnlineTest 🗸 🗸	🗉 运行测试 🛛 5					
東谷測试	用例列表	WLDTEST-iOS用例					
性能测试 🛯 2	▼	任务列表用例详情					
远程真机	じ WLDTEST-iOS用例	应用版本: 全部	✓ 选择设备: 全部	V			
测试管理 🖌 🖌							
解决方案 💙		CPU占用	内存占用	帧率		网络均值	
> 计费相关		0.00 %	Омв	0 FF	PS	О кв/s	
帮助		名称 开始时间 测试时长	CPU占用(%)	内存占用(MB)	帧率(FPS)	网络均值(KB/S)	操作
				暂无数据			
						< ±	页 1 下一页 >

## 5:在远程真机页面:

- 申请运行性能测试的设备,打开指定设备的远程真机页面。具体操作参见:申请远程真机。
- 在指定设备的远程真机页面 > 应用页签,安装待测试应用。具体操作参见:安装应用。
- 在指定设备的远程真机页面 > 性能测试页签,对应用进行性能测试。具体操作参见:实施性能测试。

## 后续操作

查看测试报告

## 6.3.3. 查看测试报告

## 前提条件

- 已申请远程真机实施性能测试,并将测试数据保存至指定用例。具体操作参见:提交测试
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

## 打开用例页面

- 1:在移动测试控制台,打开**测试产品 > 性能测试**页面。
- 2:选择已进行性能测试的应用,并在用例列表中选择用例,在右侧打开指定用例页面。

移动研发平台EMAS / 移动测试 / 性能测试						
移动测试 🛛 ₃						
测试产品 1 A CAPMOnlineTest 🗸	🛛 运行测试					
兼容测试 功能测试 用例列表 ::	WLDTEST-iOS用例					5
性能测试 III 2 ► WLDTEST-iOS目录	任务列表 用例详情					
远程真机 [] WLDTEST-iOS用例 4 测试管理 ~	应用版本: 1(0) 、 选择设	2 <b>备:</b> 全部	$\sim$			
解决方案 > 计费相关	срија радо основана се среда и се	字占用 MB	帧率 <b>14</b> .89 FPS		网络均值 О кв/S	
帮助	名称	开始时间	测试时 CPU占用 长 (%)	内存占用 帧率 (MB) (FPS	网络均值 S) (KB/S)	操作
	2020-12-02 14:43:22 APMOnlineTest(com.aliyu.emas.APMOnlineTest)	2020-12-02 14:43:25	01:56 0.26	0 14.8	9 0	详情 删除
					< 上一页	1 下一页

## 查看任务列表

## 操作说明

- 1:在指定用例页面,打开**任务列表**子页签。
- 2:输入筛选条件,查询满足条件的测试任务,及其统计信息。
- 3:在任务列表中,单击操作列的详情按钮,打开指定任务的任务详情页面。
- 4:单击操作列的删除按钮,删除指定任务。

WLDTEST-iOS用例								
任务列表 1时详情	2		8					
应用版本: 1(0)	✓ 选择设备:	8	v					
CPU占用	内存占用		破率			50	结均值	
0.26 %	Омв		14	.89 FPS		C	KB/S	
名称		开始时间	测试时 长	CPU占用 (%)	内存占用 (M®)	帧率 (FPS)	网络均值 (KB/S)	HAT.
2020-12-02 14:43:22 APMOnIneTest(com.aliyu.emas.APMOn	(ineTest)	2020-12-02 14:43:25	01:56	0.26	0	14.89	• 4	详细:删除
							< 上一页	1 T-X

## 筛选条件

选项	说明
应用版本	用于指定应用的版本。
选择设备	用于指定测试设备的品牌型号。默认为 <b>全部</b> 。

## 统计数据

选项	说明
CPU占用	指定应用版本在指定测试设备上的平均CPU占用率。
内存占用	指定应用版本在指定测试设备上的平均内存占用率。
帧率	指定应用版本在指定测试设备上的平均帧率。
网络均值	指定应用版本在指定测试设备上的平均网络传输速率。

## 查看用例详情

## 操作说明

1:在指定用例页面,打开**用例详情**子页签。

2: 输入筛选条件,查询满足条件的用例及其详情。

## 移动测试

版本: 10	0) ×	选择设备: 全部	∨ 指序: 按版:	× √	2
i本对比					
版本	平均应用CPU(%)	平均内存(MB)	平均FPS	平均网络(KB/S)	
NLL .	0.26	0	14.89	0	
1	0.26	0	14.89	0	
PU趋势					
% 0.3 m					
0.25		0			
0.2-					
0.15-					
0.1 -					

#### 筛选条件

选项	说明
应用版本	用于指定应用的版本。可以选择1个或多个应用版本,进行版本对比。
选择设备	用于指定测试设备分组,包含 <b>高端机/中端机/低端机</b> 。默认为 <b>全部</b> 。
排序	用于设置排序的方式,包括 <b>按版本 / 按创建时间</b> 。默认为 <b>按版本</b> 。

## 6.4. 远程调试

本文介绍租用特定机型复现问题,进行远程调试的操作。

## 前提条件

- 已购买远程真机资源包,或使用量未超过免费阈值。具体说明及操作请参见:产品定价。
- (可选)已上传待调试的iOS应用。具体操作请参见:上传应用。
- 已打开移动测试控制台页面。具体操作请参见: 打开控制台页面。

## 申请远程真机

1:在移动测试控制台的左侧导航栏,选择测试产品 > 远程真机页签,打开远程真机资源列表。

2:在搜索框输入关键字,例如Apple,查询指定品牌/机型的远程真机资源;或单击**展开筛选**,展开远程真机的筛选条件,指定1个或多个选项,筛选满足条件的远程真机。

### ? 说明

筛选条件包括品牌/Android版本/iOS版本/分辨率/上市时间等。

## 3:在远程真机资源列表,将鼠标移动至Apple设备标签,单击**立即使用**,打开远程真机页面。

⑦ 说明iOS真机均为非破解版。



测试服务 1	^		· 备	测试	报告
兼容测试 功能测试	3	Q. Apple 厦	研稿选 > 4	剩余	休验时长: 30:00 剩余资源时长: 00:00 ❷
性能测试 远程真机 2		Apple - iPad Pro ● 空闲	Apple - iPhone 6 ● 空闲	Apple - iPhone 5s ● 空闲	Apple - iPhone 6plus ● 空闲
隐私合规 测试管理 用例管理 任务管理 应用管理	^	<ul> <li>○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○</li></ul>	分補率:1334x750 系统版本:12.5.5 设备分级:中调机型 文明時用 5	分解率: 640x1136 系統版本: 12.5.5 设备分级: 低調机型 立即使用	<ul> <li>分補率: 1920x1080</li> <li>系統域本: 12.5.4</li> <li>设备分级: 中端机型</li> <li>文即使用</li> </ul>
符号表管理 解決方案 考察服务 考有云服务 计费相关 帮助	~	Apple - iPhone 13 pro ● 空词 分辨率: 2778x1284 系统成本: 15.0.2 设备分级: 高碘机型 文朗使用	Apple - iPhone 6s plus  全肉 分解率: 1080x1920 系統版本: 14.8 设备分级: 中調机型 立即使用	Apple - iPhone 6       全肉         分解率: 1334x750       系統成本: 12.5.5         以留分级:中請机型       文即時用	Apple - iPhone 6s     空荷       ● ● ● ●     分辨率: 750x1334       ● ● ● ●     系统成本: 14.8       ② ● ●     ② ●       ○ ● ●     ○
		Apple - iPhone 7p ● 空闲 の の の の の の の の の の の の の の の の の の の	Apple - iPhone 55       空肉         ● 第二       分辨率: 320x568         系成版本: 12.5.4       设备分级: 低清机型         ● 第二       立即使用	Apple - iPhone 12       空词         ● 資源       分辨率: 2532×1170         系成版本: 14.7.1       设备分级: 高端机型         ● 資源       立即使用	Apple - iPhone xs max ● 空闲 分辨率: 2688x1242 系成版本: 14.4.2 设备分级: 高端机型 立即使用

## 远程真机说明

远程真机用于在Web页面模拟终端设备的基本操作。



序号	按钮	控件	说明
		终端类型	远程真机的品牌、型号。
		时长	<ul><li>远程真机的体验时长或使用资源包的时长。</li><li>远程真机的剩余体验时长和剩余资源包的时长。</li></ul>
1     系统设置       近转屏幕	高清	切换设备显示屏为高清/标清显示。	
	系统设置	打开系统设置	打开系统设置页面。
		旋转屏幕	切换设备横屏/竖屏显示。
		获取设备剪贴板内容	复制远程真机剪贴板上的内容到控制台。
		快捷键盘	快捷输入窗口,支持中英文、数字、普通符号输入。

序号	按钮	控件	说明
2	全屏按钮	-	单击全屏按钮,全屏显示远程真机。
3	Home键	-	iOS终端的Home键。
4	页面调节滑块	-	用鼠标拖动页面调节滑块,调整远程真机的窗口大小。

## 安装应用

1. 在指定设备的远程真机页面,默认打开应用管理页签。

2. 查找应用。

- i. 如待调试应用已上传,单击**已上传/已安装**,筛选相应状态的应用。
- ii. 在搜索框输入关键字,模糊匹配应用的名称,查找指定应用。
- 3. 上传应用。如待调试应用未上传,单击上传应用,即刻上传,具体操作请参见:上传应用。
- 4. 安装应用。如待调测应用已上传未安装,单击操作列的安装/安装(免重签名),在远程真机上安装应用。

## ? 说明

- **安装和安装(免重签名)**的差异如下:
  - 普通安装:支持debug包,由平台进行重签名并且安装。
  - 免重签名安装:支持企业包(企业证书签名),平台直接安装。



## 查看日志

- 1:在指定设备的远程真机页面,打开日志页签。
- 2:在日志页签,开启开/关键,开始记录设备日志。
- 3:在**选择级别**下拉列表,选择日志的级别,显示指定级别的日志。
- 4: 在关键词过滤文本框, 输入关键字, 模糊匹配日志信息, 查找满足条件的日志。
- 5:开启滚动开关键,日志列表滚动显示最新的日志。
- 6:单击**下载日志**,将日志信息以文本文件的形式,下载至本地。
- 7: 单击**清除日志**,清空日志列表。

应用管理	1 日志	載屏	性能测试		更多 🗸 🏧
<b># )</b> 35	吸別 🗸	关键现法	2		✓○流动 上下戦日志 満除日
时间	PID	级别	Tag	数量	日志
15:48:04.00	57	VERBOSE	identityservicesd		No budget specified
15:48:04.00	57	VERBOSE	identityservicesd		=> Invalid URI, phone number, with no '+' cha
15:48:04.00	57	VERBOSE	identityservicesd		=> Bogus ID query for: <private> ignoring this</private>
15:48:04.00	57	VERBOSE	identityservicesd		We're at 0 queries for this hour
15:48:04.00	57	VERBOSE	identityservicesd		=+ Invalid URI, phone number, with no '+' cha

## 截屏/录屏

## 打开截屏页签

- 1:在指定设备的**远程真机**页面,打开**截屏**页签。
- 2:在截**屏**页签,包括截屏和录屏2个子页签。
- 3:选择**截屏**子页签,查看截屏列表,进行截屏操作;选择**录屏**子页签,查看录屏列表,进行录屏操作。
- 4: 单击大图/小图, 切换截屏/录屏列表预览大小。



## 截屏

- 1:选择**截屏**子页签。
- 2:在左侧设备模拟器操作应用。
- 3: 单击**点击截屏**热区, 截取设备模拟器当前界面, 并在列表显示。



## 录屏

- 1:选择**录屏**子页签。
- 2:单击**开始录屏**热区,开始录屏操作。同时,热区变为**停止录屏**。
- 3:在左侧设备模拟器操作应用。
- 4: 单击停止录屏热区,终止录屏操作,并将整个操作的录屏预览在列表中显示。



## 编辑

- 1: 将鼠标浮动到截屏/录屏预览上方,选中快捷复选框,选中指定截屏/录屏;或单击全选,选中所有截屏/录屏。
- 2:选中1个或多个截屏/录屏后,单击**下载**,下载选中的截屏/录屏;单击删除,删除选中的截屏/录屏。



## 定位测试

- 1. 在指定设备的远程真机页面,单击更多下拉菜单,选择定位,打开定位页签。
- 2. 在左侧远程真机中,打开已安装应用,进入应用的待测试页面。
- 3. 设置目标位置。
  - i. 设置**经度、纬度**,单击确定。

```
ii. 在位置搜索中输入目标位置,单击Enter。
```

## 定位成功后系统提示"虚拟定位成功"。

```
? 说明
```

- 远程真机需开启定位权限。
- 浏览器需开启定位权限,并在浏览器弹窗中单击"允许使用定位选项"。
- 请确保处于公网环境, 地图需在公网环境下显示。

移动研发平台EMAS / 移动测试 / 远程真机	DingGov 👻
Image: section of the section of th	
我的指南 + 新建指南 (D)	

## 扫码测试

- 1. 在指定设备的**远程真机**页面,单击**更多**下拉菜单,选择**扫码**,打开**扫码**页签。
- 2. 单击热区选择文件,或将文件拖动至热区,上传文件到指定目录。
- 上传成功后,系统提示**文件上传成功**。
- 3. 在左侧远程真机中,打开已<mark>安装应用</mark>,开启二维码扫描功能。
- 4. 扫描方式选择**相册**。
- 5. 在相册中选择待扫描的文件, 系统自动扫描文件。

+ AN005rova © 14:45 @ © 14:45 @ © 14:45 @		¥ (ð 100%) 📾 5-28	x 金剛	应用管理 日本 氣厚 文件 扫码 2 ← 点台或诗文件相线到此区域上传蓝色前日录,仅支持整片关型大小100M	は (1) (1) (1) (1) (1) (1) (1) (1)
				接供指導:●  ② 提示 1.上作二端简直描绘记载工地上作 2.打开应用有点二键相目描 3.选择型的注意理目描 3.选择型的注意理目描 3.选择型的注意理目标	
	< 0	0 [			8

## 实施性能测试

- 1:在指定设备的**远程真机**页面,打开**性能测试**页签。
- 2:在应用下拉列表,选择待测试的应用进程。
- 3: 单击, 开始测绘指定应用指定进程的性能数据。同时 ) 变为 。
- 4: 在左侧远程真机上操作应用, 观察测绘数据的变化。

## 곗 说明

将鼠标移动至趋势图上方,可查看指定时刻的瞬时数值。

5:单击 (a),打开保存到用例右侧栏,输入任务名称,选择保存的目录和用例,单击确定,保存测绘数据至指定目录和用例。

R存到用例	
任务名称:	
2020-12-01 15:08:26 APMOnlineTest(com.aliyu.emas.APMOnlineTest	1
R91 ©:	
	新建用例
	新建口来
暂无用例数据	
请新建用例	
新建田樹 新建田泉	

## ? 说明

目录和用例需预先设置,如未设置,单击**新建用例/新建目录**,即刻添加。新建的目录和用例,后续在**性能测试的用例列表**显示。具体操作请参见:管理测试用例。

#### 6: 单击 , 完成性能数据的测绘。





性能测试报告及相关数据说明,请参见:查看测试报告。

# 7.管理测试任务

## 前提条件

- 已提交Android/iOS应用的兼容/功能测试。具体操作请参见:实施Android/iOS测试 > 兼容/功能测试 > 提交测试。
- 已打开控制台页面。具体操作请参见:打开控制台页面。

## 操作步骤

- 1: 在移动测试控制台页面,打开测试管理 >任务管理页面,查看已提交的测试任务。
- 2:如测试任务的**状态为运行中**,单击**刷新**,查看最新信息。
- 3:在搜索框输入关键字,模糊匹配应用名/包名,筛选满足条件的测试任务。
- 4:在任务列表:
- 单击测试类型/状态列的筛选,并设置选项,筛选满足条件的测试任务。
- 单击备注列的编辑,修改指定测试任务的备注信息。
- 单击操作列的查看链接,打开指定测试任务的测试报告。具体请内容参见:实施Android/iOS测试>兼容/功能测试>测试报告说明。
- 单击操作列的分享链接,生成当前测试报告的分享链接。通过分享链接,任意用户均可使用浏览器查看该测试报告。

移动研发平台EMAS / 移动测试 / 任务管理

移动测试	3	4						
测试产品 🖌	〇 刷新	Q 搜索应用名、包名						
测试管理 1 个	编号	测试应用 5	测试类型	提测时间	測试结果 💿 🛛 🌀	状态	备注	操作
用例管理	720943	APMOnlineTest(1.3)	iOS功能测试	2020-12-02 11:19:56	10 (3/7/0)	● 已完成	∠ 8	查看   分享
应用管理	720922	APMOnlineTest(1.3)	iOS兼容性测试	2020-12-02 11:04:48	10 (8/2/0)	● 已完成	<u>~</u>	查看 分享
解决方案 🖌 🖌	720383	UlKitCatalog(13.2)	iOS兼容性测试	2020-12-01 11:44:58	2 (2/0/0)	<ul> <li>已完成</li> </ul>	2	查看 分享
计费相关	720382	爱奇艺(11.10.0)	Android兼容性测试	2020-12-01 11:41:59	3 (2/1/0)	● 已完成	<u>~</u>	查看 分享
, 帮助	719067	手机淘宝(9.15.0)	Android功能测试	2020-11-26 17:39:00	10 (3 / <mark>4 / 3</mark> )	● 已完成	WLDTEST	查看 分享
	718476	手机淘宝(9.15.0)	Android兼容性测试	2020-11-25 11:30:56	10 (6 / 4 / 0)	● 已完成	WLDTEST: Android 🤌	查看 分享
	716532	优酷视频(8.2.0)	Android功能测试	2020-11-17 20:35:56	6 (6/0/0)	● 已完成	<u>~</u>	查看 分享
	716531	爱奇艺(11.10.0)	Android兼容性测试	2020-11-17 20:33:46	3 (2/1/0)	● 已完成	2	查看   分享
	700089	TestMe(1.0)	Android兼容性测试	2020-09-08 10:14:10	1 (0/1/0)	● 已完成	<u>@</u>	查看 分享
	692540	【阿里云移动推送】 Demo程序(1.0)	Android功能测试	2020-08-05 17:06:17	10 (4/6/0)	● 已完成	<u>/</u>	查看「分享」
							< 上一	瓦 1 2 下一页 >

# 8.申请专家测试

专家测试服务,包含App兼容测试、功能测试、远程调试、隐私合规检测等测试服务,并由阿里云测试专家一对一定制测试脚本,提供专业的测试报告分析,承诺48小时 内交付测试报告。

## 前提条件

- 如涉及测试微信小程序场景,客户需自行提供测试所需微信账号。具体说明参见:常见问题。
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面。

## 打开专家测试介绍页面

在移动测试控制台页面,打开**解决方案 > 专家服务 > 专家测试介绍**页面,查看移动测试专家服务详情。

移动测试			
测试服务	~	专家测试介绍	需求管理     测试任务
测试管理	~		
解决方案	^		移动测试专家服务
专家服务			移动测试专家服务致力于提供专业、稳定、全面、高价值的APP测试能力,帮助客户以最低的成本、最高的效率发现
专有云服务			APP中的各獎總重(總私告规、APP崩凌、各獎兼举性问题、切能性问题、任能问题等)。移动测试拥有大量热门机型,要盖市场上95%的用户。专业的测试能帮助APP减少用户流失,提高APP质量和市场竞争力。
计费相关			提 <mark>交需求</mark> 」 下載报告样例 服务咨询: 钉钉搜索群号 11762195
帮助			

## 下载报告样例

在专家测试介绍页面,单击下载报告样例按钮,下载专家测试服务最终输出的报告样例。

专家测试介绍	需求管理	测试任务
<b>移动測试</b> 本PP 本PP 本PP 本PP 本PP 本PP 本PP 本P	专家服务 专家服务取力于提供专业、稳定、全面、高价值的APP则 英雄醫學(聯私合规、APP崩溃、各类兼容性问题、功能 市场上95%的用户,专业的测试能帮助APP减少用户流失 R 山下载报告样例 服务咨询:钉钉搜索群号	试能力,要助客户以最低的成本、最高的效率发现 生问题 性能问题等)。移动则试拥有大量热门机 ,提高APP质量和市场竞争力。 * 11762195

## 提交测试需求

在专家测试介绍页面,单击提交需求,打开提交专家测试需求页面。

专家测试介绍	需求管理	测试任务
移动測试专家服务 参动測试专家服务 参动測试专家服务 参动測试专家服务 参动測试专家服务 参动測试专家服务 参动测试专家服务 参动测试专家服务 参动测试专家服务 参动测试专家服务 参动测试专家服务 参动测试专家服务 参动测试专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速专家服务 参切加速重 参加加速	专业、稳定、全面、高价值的APP测试能力,帮助客户以最低的成本、最高的)。移动测试拥有大量热门机型,覆盖市场上95%的用户。专业的测试邮帮 1年例   服务咨询:钉钉搜索群号 11762195	的效率发现APP中的各类隐患(隐私合规、APP崩溃、各类兼容性问 bAPP减少用户流失,提高APP质量和市场竞争力。

#### 或在需求管理页面,单击提交需求,打开提交专家测试需求页面。

专家测试介绍	需求管理	测试任务
全部 ∨ Q 全部		
	暂无数据	
请提交需求,开启您的	专家测试之旅,体验阿里云提供的专业、稳定、全面、	高价值的APP测试能力
	提交需求	

2. 在提交专家测试需求页面,添加Android应用/iOS应用/小程序测试要求,及其他特殊测试要求。

## ○ Android应用测试要求

Android应用测试				
兼容测试				
设备台数:	请输入0-600中的数字	台		
测试时长:	10分钟 ~			
功能测试				
设备台数:	请输入0-600中的数字	台		
测试用例数:	100	$\uparrow$		
UI适配测试(	•			
◎ 添加UI适配测试				
是否需要复测	0			
🗌 添加复測				
测试次数				
请输入1-500中的数学 次				
隐私合规检测				
✔ 个人信息收集合规检测 ⊚				
✓ 智能隨私政策合規检測 ②				

×

#### 配置说明如下:

选项	说明	
兼容测试	如需进行兼容性测试,则输入执行兼容测试的 <b>设备台数</b> ,以及应用在每台设备上执行的 <b>测试时长</b> 。	
功能测试	如需进行功能测试,则输入执行功能测试的 <b>设备台数</b> ,以及用于验证应用功能的 <b>测试用例数</b> 。	
UI适配测试	Ul适配测试用于检测应用在不同设备上的页面布局/风格是否显示正常;操作/控件交互/友好性等是否体验正常。 如需进行Ul适配测试,则选中 <b>添加Ul适配测试</b> 复选框。	
是否需要复测	复测,即根据第一次测试的结果对应用进行修复后,再次由测试专家实施同类测试,进行双重确认。 如需进行复测,则选中 <b>添加复测</b> 复选框。	
测试次数	以上测试需求,可一次性购买多次。 例如,计划每月迭代一次,并进行测试,可填写12次包年;计划对1.0和2.0两个版本进行测试,可填写2次等;仅计划对当前应用进行 测试,则填写1次即可。	
隐私合规检测	隐私合规检测包括: ■ 个人信息收集合规检测:基于对apk文件的代码检测,主要包括基本信息检测、权限检测、采集行为检测和SDK检测等。 ■ 智能隐私政策合规检测:基于对隐私政策文本的检测,主要包括形式合规分析、NLP采集检测和未声明采集检测等。 ↓ 注意 智能隐私政策合规检测必须与个人信息收集合规检测结合使用。	

#### ○ iOS应用测试要求

iOS应用测试					
兼容性测试					
设备台数:	请输入0-60中的数字	台			
测试时长:	10分钟 ~				
功能测试					
设备台数:	请输入0-60中的数字	台			
测试用例数:	100	Ŷ			
UI适配测试 《	UI适配测试 💿				
□ 漆加UI适I	記測试				
是否需要复测	0				
() 添加复测					
测试次数					
请输入1-500中的数字 次					

×

选项	说明
兼容性测试	如需进行兼容性测试,则输入执行兼容测试的 <b>设备台数</b> ,以及应用在每台设备上执行的 <b>测试时长</b> 。
功能测试	如需进行功能测试,则输入执行功能测试的 <b>设备台数</b> ,以及用于验证应用功能的 <b>测试用例数</b> 。

选项	说明
UI适配测试	Ub适配测试用于检测应用在不同设备上的页面布局/风格是否显示正常;操作/控件交互/友好性等是否体验正常。 如需进行Ub适配测试,则选中 <b>添加Ub适配测试</b> 复选框。
是否需要复测	复测,即根据第一次测试的结果对应用进行修复后,再次由测试专家实施同类测试,进行双重确认。 如需进行负测,则选中 <b>添加复测</b> 复选框。
测试次数	以上测试需求,可一次性购买多次。 例如,计划每月迭代一次,并进行测试,可填写12次包年;计划对1.0和2.0两个版本进行测试,可填写2次等;仅计划对当前应用进行 测试,则填写1次即可。

#### ○ 小程序测试要求

## ↓ 注意

小程序测试需客户提供微信测试账号,具体数量由测试专家评估。具体说明参见:常见问题。

×

小程序测试			
功能测试			
Android台数	请输入0-600中的数字	台	
iOS台数:	请输入0-60中的数字	台	
测试用例数:	100	$\uparrow$	
UI适配测试(	•		
□ 漆加U适配测试			
是否需要复测 😡			
添加复测			
测试次数			
请输入1-500中的数字 次			

## 配置说明如下:

选项	说明
功能测试	如需进行功能测试,则输入执行功能测试的Android/iOS <b>设备台数</b> ,以及用于验证应用功能的 <b>测试用例数</b> 。
UI适配测试	Ul适配测试用于检测应用在不同设备上的页面布局/风格是否显示正常;操作/控件交互/友好性等是否体验正常。 如需进行Ul适配测试,则选中 <b>添加Ul适配测试</b> 复选框。
是否需要复测	复测,即根据第一次测试的结果对应用进行修复后,再次由测试专家实施同类测试,进行双重确认。 如需进行负测,则选中 <b>添加复测</b> 复选框。
测试次数	以上测试需求,可一次性购买多次。 例如,计划每月迭代一次,并进行测试,可填写12次包年;计划对1.0和2.0两个版本进行测试,可填写2次等;仅计划对当前应用进行 测试,则填写1次即可。

## ○ 特殊测试要求

特殊测试要求(选填):

- 📃 是否存在账号互踢登录?
- 📃 是否存在图片、短信验证码?
- 📃 是否协助绕过图片、短信验证码?

根据应用的实际情况,选择相应选项。

- 3. 设置期望交付时间,输入需求描述。
- 4. 可选上传待测试应用和需求文档。

🥐 说明

- 。 应用仅支持apk/ipa格式文件,且文件须小于2G。
- 需求文档可支持doc/docx/xls/xlsx/pdf格式文件,且文件须小于100M。

📃 是否存在随机密码键盘?

_______ 是否需要测试包含第三方账号登录?
5. 输入用户信息,便于后续联系沟通。

用户信息	
企业名称:	
邮箱:	
请输入邮箱,多个请以英文逗号分隔	
手机/座机(选填):	
请输入手机或座机,多个请以英文逗号分解	l.
请输入邮稿,多个请以英文显号分隔 手机/废机(这项): 请输入手机或座机,多个请以英文温号分隔	1

4. 单击提交按钮,提交测试需求;单击保存,稍后提交按钮,保存测试需求。
 需求提交后,打开需求管理页面,查看已提交的测试需求。

专家测试介绍 需求管理 测试任务				
全部 ∨ Q 全部				
🖾 提交新需求	WLDTEST 2020/12/09	详情 💽 评估中	前端测试 2020/12/03	详情 💽 评估中
+	📥 Android测试	O / 1	📥 Android测试	O / 1
「「「」「」「」」「」」「」」「」」「」」「」」「」」「」」「」」」「」」「」			iOS测试	<b>O</b> / 1
16A 11073				
	取消需求		取消需求	

# 取消测试需求

- 1:在需求管理页面,查看已提交的测试需求。
- 2: 如测试需求的状态为评估中,单击取消需求按钮,取消指定的测试需求。

专家测试介绍 需求管理 1 王务				
全部 v Q 全部				
🕒 提交新需求	WLDTEST 2020/12/09	详情 💽 评估中	2 前端测试 2020/12/03	详情   ● 评估中
+	📥 Android测试	<b>O</b> / 1	📥 Android测试	O / 1
「提交需求			🗯 iOS测试	<b>O</b> / 1
	取消需求	3	取消需求	

测试需求取消后,状态变更为**已取消**。

## 重新提交需求

- 1:在**需求管理**页面,查看已提交的测试需求。
- 2:如测试需求的状态为已取消,单击重新提交需求按钮,打开提交专家测试需求页面。

专家测试介绍 需求管理 1任务				
全部 v Q 全部				
🕒 提交新需求	WLDTEST 2020/12/09	详情 😑 已取消	2 前端测试 2020/12/03	详情 🕒 评估中
+	📥 Android测试	0 / 1	📥 Android测试	<b>O</b> / 1
提交需求			🗙 iOS测试	0 / 1
	重新提交需求	3	取消需求	

3: 在提交专家测试需求页面,修改测试需求,配置说明同:提交测试需求

# 下单并支付

- 1:在需求管理页面,查看已提交的测试需求。
- 2:如测试需求已评估完成,测试需求的状态变更为待支付。
- 3: 单击进入详情页按钮, 打开专家测试需求详情页面。

专家测试介绍	需求管理 1式任务				
全部 ∨ Q	全部				
🕒 提交新需求	十 提交需求	WLDTEST 2020/12/09 ● Android测试	详情 ● 待支付 2 qw 0 / 10	<b>vef243t34t45rg 2020/03/25</b> ▲ Android测试 矿 小程序测试	详情 ● 评估中 O / 10 O / 10
		评估已完成,请 <mark>进入详</mark>	青页 <mark>支 3</mark>	取消需求	
在 <b>专家测试需</b> 对	<b>找详情</b> 页面,查看 <b>费用</b> 和 <b>预估工时</b> 。				
基本信息					
企业名称:		测试次数:20(Andro	oid 端 10 次,小程序 端 10 次)	需求状态: ●	待支付
联系方式:				费用: ¥ 44,0	000 元(共 20 次
质估工时					
測试端	测试项			预估工时(小时)	
	UI适配测试			2	
	兼容性测试 ( 机型 600 款,10 分钟/台 )			2	
	功能测试 ( 机型 600 款,用例 100 个 )			3	
Android	深度性能测试			4	
	报告审核			6	
	复测			5	
	UI适配测试			3	
	兼容性测试 ( 机型 60 款,10 分钟/台 )			2	
iOS	功能测试 ( 机型 60 款,用例 100 个 )			3	
	报告审核			2	
	复测			1	
	UI适配测试			2	
小程度	功能测试 ( Android: 500 款,iOS: 50 款,)	用例 100 个)		1	
小性疗	报告审核			2	
	复测			2	
如无异议,单击 <b>← 专家测试</b> ^{週试产品} ~ ^{週试管理} ~	<b>下单并支付</b> 按钮,进行付款;否则, <b>式需求详情</b> 基本信息 企业名称:	单击 <b>取消需求按钮,</b> 取消当前测量 测试次数: 20 (Android 端 10	<b>式需求。</b> 2次,小程序端10次)	□ 下年 需求状态: ● 待支付 再用: V 44 000 = (# 20 %)	并支付 ————————————————————————————————————
解决方案 ^ <b>专家服务 (Hot</b> )	测试内容 处理日志			and a subsection of the particular	
专有云服务	測试内容				
计费相关	Android 端测试内容	<b>拿</b> 小程序 端测试内容			
帮助	測试次数: 10(已使用0次) 兼容性測试: 机型10款,10分钟/台 功能測试: 机型10款,用例100个	测试次数: 10(已使 功能测试: Android机	用 0 次) 型 20 款, iOS机型 0 款		

支付完成后,测试需求的状态变更为**已支付**。

# 发起测试

- 1:在需求管理页面,查看已提交的测试需求。
- 2:如测试需求的状态为已支付,单击发起测试按钮,打开发起测试右侧栏。
- 3:在**发起测试**右侧栏,填写测试需求。

# 用户指南·申请专家测试

发起测试	×
所属需求:	
优酷视频5 - 2020-07-09	
<ul> <li>(新会工法)</li> <li>(新会 2 次)</li> <li>(SSE用憲法(新会 2 次)</li> <li>小程序位用選試(新会 3 次)</li> </ul>	
需求描述:	
	4
期望交付时间:	
□ 请选择日期	
上传待测试应用(选填):	
点击上传 仅支持 apk、ipa 格式,大小 2048 MB 以内的文件上传	
上传需求文档(远填):	
点击上传 仅支持 doc、docx、xls、xlsx、pdf 格式,大小 100	
MB 以内的文件上传	

# 配置说明如下:

选项	说明
待测试端	选择1个或多个预设的Android/iOS/小程序测试需求,进行本轮测试。
需求描述	设置本轮测试的需求描述。
期望交付时间	设置本轮测试的期望交付时间。
上传待测试应用	应用仅支持apk/ipa格式文件,且文件须小于2G。
上传需求文档	需求文档可支持doc/docx/xls/xlsx/pdf格式文件,且文件须小于100M。

# 4: 单击**确定**按钮,发起测试,并等待测试完成。

# 发起测试后,可在**测试任务**页签,查看已发起的测试。

专家	测试介绍	需求管理	测试任务				
へ捜	索任务						
	测试任务				所属需求	使用测试次数	开始/结束时间
+	TEST			la finê n		Android测试:1 次	2020-12-09 16:09:34 2020-12-31 16:00:00
+	TEST			bromen		Android测试: 1 次	2020-12-09 16:08:44 2020-12-31 16:00:00
+	alibaba_ [±] 1. 测试背景	<b>专家测试_2020</b> 骨(含环境&账号)	<b>0-12-02_11:4</b> 说明) 2. 测试建议	<b>9</b> X方案 3. 功能側重点 4. 重点机型	alibaba-202006291139	Android测试: 1 次	2020-12-02 10:00:00 2020-12-05 16:00:00

# 导出测试报告

- 1:在**测试任务**页面,查看已发起的测试任务。
- 2:单击任务前的扩展按钮,展开任务详情。
- 3:如测试已完成,单击**导出报告**按钮,将测试报告下载到本地。

专家	则试介绍 需求管理 测试任务 1					
へ 捜	索任务					
	测试任务		所属	需求	使用测试次数	开始/结束时间
+	【复测】_多端测试-isv结算-复测_2020-04-17_11:2 测试	28	aliba	ba-202003041538	Android測试:0 次 iOS测试:0 次 小程序测试:0 次	2020-04-17 11:28:29 2020-04-25 16:00:00
-	<b>多端测试-isv结算-非复测_2020-04-17_11:27</b> 测试		aliba	ba-202003041538	Android測试:1 次 iOS測试:1 次 小程序测试:1 次	2020-04-17 11:27:54 2020-04-25 16:00:00
2	计划名称	计划状态	使用测试次数	测试时间		
	🛎 多端测试-isv结算-非复测_Android	待评价	Android测试次数:1 次	3 2020-04-25 16:00:00	3 导出报告 评分	
	● 多端测试-isv结算-非复测_iOS	计划完成	iOS测试次数:1 次	() 2020-04-25 16:00:00	导出报告   评分	: 非常满意
	📦 多端测试-isv结算-非复测_小程序	待评价	小程序测试次数: 1 次	3 2020-04-25 16:00:00	导出报告   评分	

# 9.咨询专有云服务

# 前提条件

已打开移动测试控制台页面。具体操作参见:打开控制台页面

# 操作说明

# 1:在移动测试控制台,打开**解决方案 > 专有云服务**页面,查看YunBox专有云产品详情。

移动研发平台EMAS / 移动测试 / 专有云服务

移动测试			
测试产品 > 测试管理 > 解决方案 1 ^	YunBox 专有云 —	一 部 署 在 您 公 司 的 移 动 测 试 平 台	<b>服务咨询: 钉钉搜索群号 11762195</b>
专家服务 1105	[2] 标准版	😌 专业版	1111 中央机房版
计费相关 帮助	提供软件与硬件一体化服务、可且送20台手机,提供兼 容、功能、性能、远程真机相用、在线脚本录制等服务能力	提供软件与硬件一体化服务,可自选50台手机,提供兼 容、功能、性能、远程真机租用、在线脚本录制等服务能力	提供软件与硬件一体化服务,不限手机数量、支持分布式部 署,提供兼容、功能、性能、远程真机租用等服务能力
>	成熟的专有云服务体系		
		专有云服务体系	
	<ul> <li>自动化测试能力</li> <li>兼容 功能 性能</li> </ul>	二次开发赋能企业	
	提供了常见的兼容、功能、性能等自动化测试能力。	可以基于自身业务场景需要定	- 則則這步襲。回时,我们提供了宗幣的API接口,方便企业将测试服务
	<ul> <li>● 远程真机共享</li> <li>业务部/□A 业务部/□B</li> <li>真机远程共享</li> <li>● 西部(□</li> </ul>	· 如金玉 二 4 五 2 初 两 南 美 2 元 整体 接入到服务中台。	anning an traditional anna an 1984) i bhladh 1989 anna 2

2: 联系技术支持,加入钉钉群:【移动测试】技术支持群-应用研发平台EMAS(ID: 11762195),咨询购买。

# 10.查看计费

# 前提条件

- 已注册/登录阿里云账号。具体操作参见:账号管理
- 已打开移动测试控制台页面。具体操作参见:打开控制台页面

# 操作步骤

在移动测试的控制台页面,打开**计费相关**页签。

#### 移动研发平台EMAS / 移动测试 / 计费相关 移动测试 测试产品 ~ 费用导航 测试管理 ~ 🔄 购买资源包 我的资源包 ● 费用中心 解决方案 ~ 计费相关 帮助 价格 价格说明地址

在**计费相关**页签:

- 单击**费用中心**,打开**阿里云用户中心**首页,查看当前账户的消费记录。
- 单击我的资源包,打开阿里云用户中心>资源包管理页面,查看资源包概览及其明细。
- 单击价格说明地址,打开产品定价页面,查看移动测试的产品价格。
- 单击基础测试资源包,打开EMAS-移动测试基础测试资源包页面,购买资源包。基础测试资源包可用于兼容测试和功能测试。
- 单击远程真机资源包,打开EMAS-移动测试远程测试资源包页面,购买资源包。远程真机资源包可用于远程真机、性能测试、在线录制。
- 移动测试的购买页面还包括测试专家服务,您可根据实际测试需求,购买相应资源包。具体说明参见:产品定价。

### EMAS-移动测试基础测试资源包

	套餐	移动推送	HTTPDNS	移动热修复	移动基础测试	移动远程测试	测试专家服务	测试用例编写	移动监控
基本配置	资源包类型 资源包规格	基础测试           可用于Android兼容计           (1) iOS测试一台次           (2) Android挪试一           (3) 一台次,在兼評           (4) 一台次,在助前           2台次           资源数量越多,单价	<ol> <li>計測试、Android功能測试 (, 消耗此資源包2台次; 台次, 消耗此資源包1台支 営性測试中最多10分钟, 能性測试中最多30分钟,</li> <li>100台次 越便宣</li> </ol>	、iOS兼容性测试、iOS 次; 即测试最多10分钟后, · 即测试最多30分钟后, · 1000台次	3功能测试。 一台次资源包即被消耗完; 一台次资源包即被消耗完。 10000台次			当前配置	基础测试 2台次 1 1年 加入购物车
购买量	购买数量 有效期	1 1年 资源包购买成功后即	刻生效,资源包使用后乘	余流量不支持退订					

# 11.授权RAM用户

移动测试服务可通过主账号或RAM子账号进行访问控制,如果您需要通过RAM子账号实现移动测试服务的相关功能,需要通过主账号授权并赋予子账号相应的权限策略, 以实现相关操作。

# 前提条件

- 已注册/登录阿里云主账号。具体操作参见: 账号管理。
- 已开通移动测试服务。具体操作参见:移动研发平台 EMAS > 快速入门。
- 已打开阿里云控制台首页。

# 操作步骤

# 打开RAM访问控制台页面

- 1. 在阿里云控制台首页,鼠标移动至右上角头像处,打开快捷菜单。
- 2. 在快捷菜单中,选择**访问控制**,进入**RAM访问控制**页面。



#### 创建主体

您可基于用户/用户组/RAM角色任一主体进行访问控制,请根据实际需求进行相应设置。

在**RAM访问控制**页面,执行如下操作:

- 创建用户。具体内容参见:创建RAM用户。
- 创建用户组。具体内容参见:创建用户组。
- 创建RAM角色。具体内容参见: RAM角色概览。

授权

### 在RAM访问控制页面,选择权限管理>授权,打开授权页面。

2. 在授权页面,单击新增授权,打开添加权限右侧栏。



3. 在添加权限右侧栏, 执行如下操作:

	and a second			
授权应用范围				
● 整个云账号				
) 指定资源组				
请选择或输入资源组名称进行按照	t in the second s			
<b>被抵权主法</b>				
	analiyun.com X			
(1) (人上市)       第23年(1)(日本)       第23年(1)(日本)       透描(2)(日本)       透描(2)(日本)       夏(2)(日本)       夏(2)(日本)       夏(2)(日本)	+ 新建収得編集		已透緩 (4)	清雪
alice for a for	+ MiltOffigue	1	日远譯 (4) AliyunMobileTestingReadOnlyAccess	清 ×
RUSHULT RUSHRUKULSA 透理収録 系統集略 AllyunMobileTesting 权限機能名称		1	已退降 (4) AljyunMobieTestingReadOnlyAccess AljyunMobieTestingFulAccess	清 × ×
AllyunMobileTestingReadOnL.		3	已退降 (4) AljunMobieTestingReadOnlyAccess AljunMobieTestingFulAccess AljunMhubFulAccess	潮 × × ×
AllyunMabileTestingReadOnL- AlyunMabileTestingReadOnL- AlyunMabileTestingReadOnL-		3	已送降 (4) AllyunMobieTestingReadOnlyAccess AllyunMobieTestingFullAccess AllyunMhubFullAccess AllyunMhubFeadOnlyAccess	漸 × × × ×

# 配置说明如下:

选项	操作说明
授权范围	选择云账号全部资源。
被授权主体	输入关键字,筛选并指定已创建的用户/用户组/RAM角。
选择权限	<ul> <li>i. 选择系统策略/自定义策略选项。</li> <li>ii. 输入关键字,搜索移动测试和移动研发平台(EMAS)的系统策略,包括: <ul> <li>AliyunMHubFullAccess:管理移动研发平台(EMAS)的权限。</li> <li>AliyunMHubReadOnlyAccess: 管理移动研发平台(EMAS)的权限。</li> <li>AliyunMobileTestingFullAccess:管理移动测试(MobileTesting)的权限。</li> <li>AliyunMobileTestingReadOnlyAccess: 只读访问移动测试(MobileTesting)的权限。</li> <li>自定义策略包括已添加的用户列表读权限。具体操作参见:新建用户列表读权限。</li> </ul> </li> <li>ii. 根据实际需要,选择移动研发平台(EMAS)和移动测试的系统策略,添加至右侧已选择列表。</li> <li>ジ 注意 <ul> <li>移动研发平台(EMAS)的系统策略至少选择一个。</li> </ul> </li> </ul>
	• 移动测试的系统策略至少选择一个。

4. 单击**确定**,保存设置。

# 相关内容

RAM访问控制为阿里云通用服务。如需了解更多内容,参见:访问控制。

# 12.日志解释与解决方案

# 12.1. 崩溃时错误

# 12.1.1. and roid_content_ActivityNotFoundException

# 问题分析

无法找到对应的Activity时抛出此类异常。

# 解决方案

无法找到对应的Activity时抛出此类异常,一般发生在 AndroidManifest.xml 文件没有配置需要的Activity路径所导致的或手机中确实无此应用,建议检查 AndroidManifest文件Activity路径并在启动Activity处捕获异常。

# 代码示例

android.content.ActivityNotFoundException: Unable to find explicit activity class {*}; have you declared this activity in your AndroidManifest.xm 1?

# 方案1:确认AndroidManifest.xml中所需Activity路径正确

```
<activity
android:name="com.your.package.name.YourActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
```

# 方案2:针对可能不存在的Activity抛出的异常进行捕获

```
public static void startDialer(Context context, String phoneNumber) {
    try {
        Intent dial = new Intent();
        dial.setAction(Intent.ACTION_DIAL);
        dial.setData(Uri.parse("tel:" + phoneNumber));
        context.startActivity(dial);
    } catch (ActivityNotFoundException ex) {
        Log.e(TAG, "Error starting phone dialer intent.", ex);
        Toast.makeText(context, "Sorry, we couldn't find any app to place a phone call!",
        Toast.LENGTH_SHORT).show();
    }
}
```

# 参考文献

1.https://developer.android.com/reference/android/content/ActivityNotFoundException.html 2.http://stackoverflow.com/questions/15825081/error-default-activity-not-found 3.http://www.cnblogs.com/Cjch/archive/2013/05/28/3104561.html 4.https://developer.android.com/guide/topics/manifest/manifest-intro.html(应用清单文件) 5.https://developer.android.com/guide/topics/manifest/activity-element.html(activity语法)

# 12.1.2. android_content_res_Resources_NotFoundException

# 问题分析

当请求资源却无法找到资源时会抛出此类问题。一般是因为编译器编译程序时可以找到的资源,但在运行时却找不到时会抛出此类异常。

### 解决方案

在编译器编译程序时可以找到的资源,但在运行时却找不到时会抛出此类异常。建议 clean project ,删除R文件重新build,并检查引用的资源是否正确。

# 参考文献

- https://developer.android.com/reference/android/content/res/Resources.NotFoundException.html
- http://stackoverflow.com/questions/9161435/getting-android-content-res-resourcesnotfoundexception-exception-even-when-the

# 12.1.3. android_database_sqlite_DatabaseObjectNotClosedException

# 问题分析

在打开数据库操作之前,应用程序没有关闭游标或数据库对象。

# 解决方案

# 应用程序没有关闭游标或数据库对象导致抛出此类异常。通过对游标及数据库对之前打开的数据库对象调用close方法进行关闭操作即可。

#### 代码示例

```
android.database.sqlite.DatabaseObjectNotClosedException:Application did not close the cursor or database object that was opened here
android.database.sqlite.DatabaseObjectNotClosedException:Application did not close the cursor or database object that was opened here
at android.content.ContentResolver.query(ContentResolver.java:399)
at android.content.ContentResolver.query(ContentResolver.java:316)
```

# 操作步骤

#### 1: 使用 try catch 块捕获异常,并在finally部分关闭数据库。

```
OpenHelper dh = new DatabaseHelper(this);
SQLiteDatabase db = null;
try {
    db = dh.getWritableDatabase();
} catch(SQLiteException e) {
    e.printStackTrace();
} finally {
    if(db != null && db.isOpen())
        db.close();
}
```

2:在Activity生命周期中, onResume() 打开数据库, onPause() 关闭数据库。

```
OpenHelper dh = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    dh = new DatabaseHelper(this);
}
@Override
protected void onResume() {
    this.db = dh.getWritableDatabase();
}
@Override
protected void onPause( {
    if(db != null && db.isOpen()) {
        db.close();
    }
}
```

# 参考文献

- http://stackoverflow.com/questions/7492006/databaseobjectnotclosedexception
- http://stackoverflow.com/questions/4734886/how-can-one-avoid-databaseobjectnotclosedexception
- http://debuglog.iteye.com/blog/1404338

# 12.1.4. android_database_sqlite_SQLiteCantOpenDatabaseException

### 问题分析

SQLite无法打开数据库时抛出此类异常。

# 解决方案

SQLite无法打开数据库抛出此类异常是因为读写的数据库文件不存在或者没有申请到正确的读写权限或者这个文件的权限比较特别,建议检查是否申请到了正确的权限、 并且文件路径正确。

#### 代码示例

如果需要读取外部存储器上的文件需要在Manifest添加权限:

<uses-permissionandroid:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission> <uses-permissionandroid:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>

在Android6.0以前申请权限时,用户在安装和更新时会进行授权。Android6.0以后会在运行期用户进行授权。为了在运行期获取权限,首先需要向用户请求权限:

```
// Here, thisActivity is the current activity
if (ContextCompat.checkSelfPermission(thisActivity,
                Manifest.permission.READ CONTACTS)
        != PackageManager.PERMISSION GRANTED) {
    // Should we show an explanation?
    if (ActivityCompat.shouldShowRequestPermissionRationale(thisActivity,
            Manifest.permission.READ_CONTACTS)) {
        // Show an expanation to the user <code>*asynchronously*</code> -- don't block
        // this thread waiting for the user's response! After the user
        \ensuremath{{\prime}}\xspace // sees the explanation, try again to request the permission.
    } else {
        // No explanation needed, we can request the permission.
        ActivityCompat.requestPermissions(thisActivity,
                new String[]{Manifest.permission.READ_CONTACTS},
                MY_PERMISSIONS_REQUEST_READ_CONTACTS);
        // MY_PERMISSIONS_REQUEST_READ_CONTACTS is an
        // app-defined int constant. The callback method gets the
        // result of the request.
    }
}
```

# 然后确认是否权限通过,做业务处理。

#### @Override

```
public void onRequestPermissionsResult(int requestCode, String permissions[], int[] grantResults) {
    switch (requestCode) {
        case WRITE_REQUEST_CODE:
            if(grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                //Permission granted.
                //Continue with writing files...
            }
        else{
                //Permission denied.
            }
        break;
    }
}
```

# 参考文献

- https://developer.android.com/reference/android/database/sqlite/SQLiteCantOpenDatabaseException.html
- http://stackoverflow.com/questions/17034511/android-database-sqlite-sqlitecantopendatabaseexception-unknown-error-code-14
- https://developer.android.com/training/permissions/requesting.html#perm-check (安卓6.0授权检查方式)
- https://developer.android.com/guide/topics/security/permissions.html#normal-dangerous(普通权限及危险权限说明)

# 12.1.5. android_os_DeadObjectException

#### 问题分析

调用的对象不存在,因为它所在App进程不存在或进程崩溃,此时在底层回调时报错。

# 解决方案

在调用对象之前,建议检查是否存在此进程。DeadObjectException异常出现是因为App进程不存在或进程崩溃因此在底层回调时报错。进程不存在或者进程崩溃需要查 看其他日志来定位。对于进程崩溃的情况,也有可能部分原因是由于操作系统考虑到内存、CPU、优先级等指标,选择杀死一个进程得到资源。这种情况下,可以使用 MQC平台的深度性能测试产品来帮助分析内存泄漏等问题。

# 代码示例

# 参考文献

- https://developer.android.com/reference/android/os/DeadObjectException.html
- http://stackoverflow.com/questions/6039398/android-get-pid-of-other-applications

# 12.1.6. android_os_TransactionTooLargeException

### 问题分析

Binder传输的数据存在Parcel类中,如果Binder的参数或返回值太大(超过1MB),不适合的事务缓冲区,调用会失败,并抛出TransactionTooLargeException。

# 解决方案

Binder的参数或返回值太大导致调用失败,并抛出TransactionTooLargeException,建议不要将大量数据传入Binder,避免传入大数组、大量字符串或位图。

### 代码示例

```
android.os.TransactionTooLargeException
    at android.os.BinderProxy.transact(Native Method)
     {\tt at com.android.internal.view.IInputMethodManager\$Stub\$Proxy.startInput(IInputMethodManager.java:604) } \\
   at android.view.inputmethod.InputMethodManager.startInputInner(InputMethodManager.java:1173)
   at android.view.inputmethod.InputMethodManager.checkFocus(InputMethodManager.java:1282)
   at android.view.ViewRootImpl$ViewRootHandler.handleMessage(ViewRootImpl.java:3201)
   at android.os.Handler.dispatchMessage(Handler.java:102)
   at android.os.Looper.loop(Looper.java:136)
   at android.app.ActivityThread.main(ActivityThread.java:5017)
   at java.lang.reflect.Method.invokeNative(Native Method)
   at java.lang.reflect.Method.invoke(Method.java:515)
   at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:779)
   at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:595)
    at dalvik.system.NativeStart.main(Native Method)
public class MainActivity extends Activity {
   private IDictionaryManager mDictionaryManager;
    @Override
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Intent intent = new Intent();
       intent.setAction("android.intent.action.DictionaryManagerService");
        intent.setPackage("com.wanginbeijing.dictionaryserver");
       bindService(intent, mConnection, Context.BIND_AUTO_CREATE);
        //添加一个新单词
        findViewById(R.id.btn_add).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
               try {
                   mDictionaryManager.add("你好", "Hello");
               } catch (RemoteException e) {
                   e.printStackTrace();
           }
        });
        //查询单词
        findViewById(R.id.btn_query).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
               try {
                   String english = mDictionaryManager.query("你好");
                   Toast.makeText(MainActivity.this, english, Toast.LENGTH_SHORT).show();
               } catch (RemoteException e) {
                    e.printStackTrace();
               }
           }
   private ServiceConnection mConnection = new ServiceConnection() {
        @Override
       public void onServiceConnected(ComponentName componentName, IBinder iBinder) {
            IDictionaryManager dictionaryManager = IDictionaryManager.Stub.asInterface(iBinder);
            try {
               mDictionaryManager = dictionaryManager;
               Toast.makeText(MainActivity.this, "connect success", Toast.LENGTH SHORT).show();
            } catch (Exception e) {
               Toast.makeText(MainActivity.this, "connect failed", Toast.LENGTH_SHORT).show();
                e.printStackTrace();
        1
        @Override
        public void onServiceDisconnected(ComponentName name) {
   };
}
```

# 参考文献

- https://developer.android.com/reference/android/os/TransactionTooLargeException.html
- http://stackoverflow.com/questions/11451393/what-to-do-on-transactiontoolargeexception
- https://halfstackdeveloper.github.io/2016/09/10/Android-Binder%E8%A7%A3%E5%AF%86/

# 12.1.7.

# $com_android_internal_os_RuntimeInit_UncaughtHandler_uncaughtException$

# 问题分析

此类异常发生在应用创建了线程,如果线程因为异常未捕获而退出,在Runt imelnit 中会捕获此类异常并杀死进程。具体异常的原因需要看日志附近其他log确定触发异常 信息。

# 解决方案

因为应用开启的线程抛出了异常未被捕获而导致抛出uncaughtException,建议在线程内捕获异常并做处理,如果无法在线程内捕获异常,设置线程默认的 UncaughtException Handler来处理。

# 代码示例

```
public static void main(String[] args) {
 UncaughtExceptionHandler eh=new UncaughtExceptionHandler() {
   @Override
   public void uncaughtException(Thread t, Throwable e) {
         System.out.println("uncaught Caught " + e);
  }
 };
 Thread thread=new Thread() {
   @Override
   public void run(){
     try {
       // do something and happen a exception
       throw new IOException("e");
     } catch (IOException e) {
       System.out.println("catching IOException:");
       e.printStackTrace();
     ^{\prime\prime} code may cause some other exception, we could catch these, or use setDefaultUncaughtExceptionHandler
     System.out.println(1/0);
   1
 };
 thread.setDefaultUncaughtExceptionHandler(eh);
 thread.start();
```

# 参考文献

- http://androidxref.com/4.3_r2.1/xref/frameworks/base/core/java/com/android/internal/os/RuntimeInit.java
- http://docs.oracle.com/javase/8/docs/api/java/lang/Thread.UncaughtExceptionHandler.html
- http://stackoverflow.com/questions/6546193/how-to-catch-an-exception-from-a-thread
- http://www.cnblogs.com/freeliver54/archive/2011/10/17/2215423.html

# 12.1.8. java_io_EOFException

# 问题分析

当输入过程中意外到达文件或流的末尾时,抛出此异常。此异常主要被数据输入流用来表明到达流的末尾。

### 解决方案

因为不知道流的末尾,当到达末尾的时候,抛出了此异常。这种异常主要被数据输入流用来表明到达流的末尾,建议捕获异常并退出读取输入流。

```
例如:
```

java.io.EOFException

- at libcore.io.Streams.readAsciiLine(Streams.java:203)
- at libcore.net.http.HttpEngine.readResponseHeaders(HttpEngine.java:579)
- at libcore.net.http.HttpEngine.readResponse(HttpEngine.java:827)
- at libcore.net.http.HttpURLConnectionImpl.getResponse(HttpURLConnectionImpl.java:283) at libcore.net.http.HttpURLConnectionImpl.getInputStream(HttpURLConnectionImpl.java:177)

# 代码示例

<pre>public static void main(String[] args) {</pre>
DataInputStream input = null;
try {
//do something
// Read all characters, until an EOFException is thrown.
<pre>input = new DataInputStream(new FileInputStream(FILENAME));</pre>
while(true) {
char num;
try {
<pre>num = input.readChar();</pre>
System.out.println("Reading from file: " + num);
}
catch (EOFException ex1) {
break; //EOF reached.
}
catch (IOException ex2) {
System.err.println("An IOException was caught: " + ex2.getMessage());
<pre>ex2.printStackTrace();</pre>
}
}
}
catch (IOException ex) {
<pre>System.err.println("An IOException was caught: " + ex.getMessage());</pre>
<pre>ex.printStackTrace();</pre>
}
finally {
try {
// Close the input stream.
input.close();
}
catch(IOException ex) {
System.err.println("An IOException was caught: " + ex.getMessage());
<pre>ex.printStackTrace();</pre>
}
}
1

# 参考文献

- https://docs.oracle.com/javase/7/docs/api/java/io/EOFException.html
- https://examples.javacodegeeks.com/java-basics/exceptions/java-io-eofexception-how-to-solve-eofexception/

# 12.1.9. java_io_FileNotFoundException

# 问题分析

文件未找到或无权限时抛出此类异常。

# **解决方案** 文件未找到或无权限时,请确认文件路径文件存在,添加所需访问权限,确保文件存在的情况下进行读写,并增加try catch块做异常处理。

# 代码示例

### 增加读写外部存储上的文件权限:

<uses-permissionandroid:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission> <uses-permissionandroid:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>

#### 创建文件,增加读写权限并catch exception。

示例:

java.io.FileNotFoundException: /proc/mtprof/status: open failed: ENOENT (No such file or directory)

- at libcore.io.IoBridge.open(IoBridge.java:496)
- at java.io.FileInputStream.<init>(FileInputStream.java:76) at java.io.FileInputStream.<init>(FileInputStream.java:103)
- at android.app.ActivityThread\$H.handleMessage(ActivityThread.java:1494)
- at android.os.Handler.dispatchMessage(Handler.java:111)
- at android.os.Looper.loop(Looper.java:194)
- at android.app.ActivityThread.main(ActivityThread.java:5692) at java.lang.reflect.Method.invoke(Native Method)
- at java.lang.reflect.Method.invoke(Method.java:372)
- at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:959)
- at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:754)
- Caused by: android.system.ErrnoException: open failed: ENOENT (No such file or directory)
  - at libcore.io.Posix.open(Native Method)
  - at libcore.io.BlockGuardOs.open(BlockGuardOs.java:186)
  - at libcore.io.IoBridge.open(IoBridge.java:482)
- ... 10 more

#### 异常导致应用无法启动。日志如下:

```
10-20 09:43:08.073 820 820 E WifiTrafficPoller: TRAFFIC_STATS_POLL true Token 1138 num clients 12
10-20 09:43:08.074 820 820 E WifiTrafficPoller: packet count Tx=237196 Rx=493756
10-20 09:43:08.365 820 842 W BroadcastQueue: Skipping deliver [background] BroadcastRecord{9a089ad u-1 android.net.conn.CONNECTIVITY_CHANGE}
to ReceiverList{2clb84e2 23189 (unknown name)/2000/u-1 remote:2663fdd7}: process crashing
10-20 09:43:08.401 23079 23079 E ActivityThread: mtprof entry can not be found
10-20 09:43:08.401 23079 E ActivityThread: java.io.FileNotFoundException: /proc/mtprof/status: open failed: ENOENT (No such file or directo
ry)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at libcore.io.IoBridge.open(IoBridge.java:496)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at java.io.FileInputStream.<init>(FileInputStream.java:76)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at java.io.FileInputStream.<init>(FileInputStream.java:103)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1494)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at android.os.Handler.dispatchMessage(Handler.java:111)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at android.os.Looper.loop(Looper.java:194)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at android.app.ActivityThread.main(ActivityThread.java:5692)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at java.lang.reflect.Method.invoke(Native Method)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at java.lang.reflect.Method.invoke(Method.java:372)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:959)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:754)
10-20 09:43:08.401 23079 23079 E ActivityThread: Caused by: android.system.ErrnoException: open failed: ENOENT (No such file or directory)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at libcore.io.Posix.open(Native Method)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at libcore.io.BlockGuardOs.open(BlockGuardOs.java:186)
10-20 09:43:08.401 23079 23079 E ActivityThread:
                                                     at libcore.io.IoBridge.open(IoBridge.java:482)
                                                     ... 10 more
10-20 09:43:08.401 23079 23079 E ActivityThread:
10-20 09:43:08.415 23079 23079 W Launcher: setApplicationContext called twice!
10-20 09:43:08.422 23079 23079 E Fatal exception: :java.lang.RuntimeException: Unable to start activity ComponentInfo{}: java.io.IOException: bro
ken file descriptor
10-20 09:43:08.422 23079 23079 W System.err: java.lang.RuntimeException: Unable to start activity ComponentInfo{}: java.io.IOException: broken fi
le descriptor
10-20 09:43:08.422 23079 23079 W System.err:
                                                 at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2581)
10-20 09:43:08.422 23079 23079 W System.err:
                                                 at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2656)
10-20 09:43:08.422 23079 23079 W System.err:
                                                 at android.app.ActivityThread.access$800(ActivityThread.java:178)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1512)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.os.Handler.dispatchMessage(Handler.java:111)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.os.Looper.loop(Looper.java:194)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.app.ActivityThread.main(ActivityThread.java:5692)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at java.lang.reflect.Method.invoke(Native Method)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at java.lang.reflect.Method.invoke(Method.java:372)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:959)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:754)
10-20 09:43:08.423 23079 23079 W System.err: Caused by: java.io.IOException: broken file descriptor
10-20 09:43:08.423 23079 23079 W System.err: at android.graphics.BitmapFactory.nativeDecodeFileDescriptor(Native Method)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.graphics.BitmapFactory.decodeFileDescriptor(BitmapFactory.java:714)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.app.WallpaperManager$Globals.getCurrentWallpaperLocked(WallpaperManager.java:308)
10-20 09:43:08.423 23079 23079 W System.err:
                                                 at android.app.WallpaperManager$Globals.peekWallpaperBitmap(WallpaperManager.java:270)
10-20 09:43:08.423 23079 23079 W System.err:
                                                at android.app.WallpaperManager.getDrawable(WallpaperManager.java:413)
```

更换手机桌面应用布局界面时调用native方法decodeFileDescriptor读取文件/proc/stprof/status抛出了异常未捕获导致进程崩溃,建议按照示例一判断文件是否存在并 对异常进行捕获处理。

#### 参考文献

https://developer.android.com/guide/topics/data/data-storage.html#filesExternal

http://www.cnblogs.com/feisky/archive/2011/01/05/1926177.html

# 12.1.10. java_io_IOException

#### 问题分析

IO发生异常时抛出,引起的原因很多,如相关文件不存在、无权限读写、编码错误、文件读到末尾、IO中断、JSON、URL等格式不正确等。

#### 解决方案

引起IO异常的原因很多,如相关文件不存在、无权限读写、编码错误、文件读到末尾、IO中断、JSON、URL等格式不正确等。建议捕获IO异常并做相应的业务处理。

# 代码示例

# 示例一

java.io.IOException: open failed: EACCES (Permission denied)

#### 确认文件路径,增加读写外部存储上的文件权限:

<uses-permissionandroid:name="android.permission.READ_EXTERNAL_STORAGE"></uses-permission>
<uses-permissionandroid:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission.WRITE_EXTERNAL_STORAGE"></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-permission></uses-perm

### 示例二

java.io.IOException: open failed: ENOENT (No such file or directory)

#### 增加try catch块做异常处理。

```
public void readFile(String filePath) {
    BufferedReader br = null;
    try {
       String sCurrentLine;
       br = new BufferedReader(new FileReader(filePath));
       System.out.println(sCurrentLine);
}
       while ((sCurrentLine = br.readLine()) != null) {
    } catch (IOException e) {
       e.printStackTrace();
    } finally {
       try {
           if (br != null)
               br.close();
       } catch (IOException ex) {
           ex.printStackTrace();
       }
   }
}
```

# 参考文献

- https://developer.android.com/training/permissions/index.html (权限使用文档)
- https://developer.android.com/reference/android/Manifest.permission.html (权限分类文档)
- https://www.mkyong.com/java/how-to-read-file-from-java-bufferedreader-example/
- https://developer.android.com/reference/java/io/IOException.html
- https://developer.android.com/guide/topics/security/permissions.html#normal-dangerous (普通权限及危险权限)
- https://developer.android.com/training/permissions/requesting.html#perm-check (权限授权方式)

# 12.1.11. java_lang_ArrayIndexOutOfBoundsException

### 问题分析

数组索引越界,引用的对象超出了数组的大小。

# 解决方案

```
数组索引越界,引用的对象超出了数组的大小。因此在操作数组之前查看数组大小,判断对象是否存在,如果存在则返回,否则返回null。
```

#### 示例代码

```
public String arrayOutOfBounds(String[] array, int index){
    if(array!=null && array.length>index && index>=0){
        System.out.println("content is: "+array[index]);
        return array[index];
    }
    return null;
}
```

### 详细例子:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3 at com.alibaba.mgc.test.Test.arrayOutOfBounds(Test.java:52) at com.alibaba.mgc.test.Test.main(Test.java:31)
```

这种Crash是数组本身大小为0而代码中获取了数组的对象而导致的。在SourceFile文件第86行对数组进行get操作时index越界导致的。

#### 参考文献

• https://developer.android.com/reference/java/lang/IndexOutOfBoundsException.html

# 12.1.12. java_lang_ArrayStoreException

# 问题分析

这是数组存储异常,当试图将类型不兼容类型的对象存入一个 Object[] 数组时将引发异常。

# 解决方案

当试图将类型不兼容类型的对象存入一个 Object[] 数组时将引发异常,建议根据堆栈信息修改存储对象类型。

# 代码示例

```
class Father{}
class Son extends Father{}
public void handleArrayStoreException(){
    Father[] fathers=new Son[3];
     // 此处会抛异常
    try {
       // 无法存父类对象,只能存子类对象
    fathers[0]=new Father();
   } catch (ArrayStoreException e) {
     e.printStackTrace();
   }
     // 正确的代码
    Father father=new Father();
     Son son=new Son();
    if(son instanceof Son){
         // 可以存进数组
      fathers[1]=son;
     if(father instanceof Son){
          // 无法存进数组
       fathers[2]=father;
     }
   }
```

# 参考文献

• https://docs.oracle.com/javase/7/docs/api/java/lang/ArrayStoreException.html

# 12.1.13. java_lang_ClassCastException

# 问题分析

在进行强制类型转换时,转换的对象不是转换的类型的实例而抛出此类异常。

### 解决方案

在进行强制类型转换时,转换的对象不是转换的类型的实例而抛出此类异常。可以通过 if (object instance of 类型) 的语句来判断object的类型是否可强制类型转换来 进行。

# 代码示例

```
public Son getSon(Father reference) {
    if(reference instanceof Son) {
        Son son=(Son)reference;
        return son;
    }
    return null;
}
```

# 参考文献

• https://docs.oracle.com/javase/7/docs/api/java/lang/ClassCastException.html

# 12.1.14. java_lang_ClassNotFoundException

# 问题分析

ClassNotFoundException是通过 Class.forName(java.lang.String) 等方式试图通过String找到类而未找到的报错,这种错误只能在运行期抛出。

# 解决方案

通过 Class.forName(java.lang.String) 等方式试图通过String找到类而未找到的报错,这种错误只能在运行期抛出,请确认类路径正确并已正确引用。

# 代码示例

package com.alibaba.mqc.test; import java.lang.reflect.InvocationTargetException; import java.lang.reflect.Method; // @ fublic class V { public class V { public void printf() { System.out.println("printf"); } public static void main(String[] args) throws ClassNotFoundException, SecurityException, NoSuchMethodException, IllegalArgumentException, IllegalAccessException, InvocationTargetException, InstantiationException { Class c = Class.forName("com.alibaba.mqc.test.V"); Method m = c.getDeclaredMethod("printf",null); m.invoke(c.newInstance(),null); } } }

# 参考文献

• https://docs.oracle.com/javase/7/docs/api/java/lang/ClassNotFoundException.html

```
http://www.cnblogs.com/xing901022/p/4185514.html
```

# 12.1.15. java_lang_IllegalArgumentException

### 问题分析

根据Android官方文档,当方法接收到了一个不合法或不正确的参数时会抛出此类异常。

#### 解决方案

因为参数不合法导致抛出llegalArgumentException,请根据堆栈信息描述的出错的函数参数,按照要求传入正确的参数。

# 代码示例

```
public static void getUserAgent(Context context) {
  WebView webview = new WebView(context);
  WebSettings settings = webview.getSettings();
  userAgent = settings.getUserAgentString();
  Log.i("UserAgent: " + userAgent);
}
```

### 示例一

java.lang.IllegalArgumentException: HTTP parameters may not be null

- at org.apache.http.params.HttpProtocolParams.getUserAgent(HttpProtocolParams.java:150) at org.apache.http.impl.client.AbstractHttpClient.isMOMMS(AbstractHttpClient.java:790)
- at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java.56) at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java.563)
- at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:520)
- at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:498)
- at com.g.utils.HttpClientUtil.post(HttpClientUtil.java:99)
- at com.g.utils.LoadDataFromServer\$2.run(LoadDataFromServer.java:195)

#### 因为参数为空导致获取字符集失败抛出的异常。

在该示例中,getUserAgent因为userAgent变量为空,所以报错。User Agent中文名为用户代理,简称UA,它是一个特殊字符串头,使得服务器能够识别客户使用的操 作系统及版本、CPU类型、浏览器及版本、浏览器渲染引擎、浏览器语言、浏览器插件等。

#### 示例二

java.lang.IllegalArgumentException

- at android.view.Surface.nativeLockCanvas(Native Method)
- at android.view.Surface.lockCanvas(Surface.java:243)
- at android.view.ViewRootImpl.drawSoftware(ViewRootImpl.java:2435)
- at android.view.ViewRootImpl.draw(ViewRootImpl.java:2409)
- at android.view.ViewRootImpl.performDraw(ViewRootImpl.java:2253)
- at android.view.ViewRootImpl.performTraversals(ViewRootImpl.java:1883)
- at android.view.ViewRootImpl.doTraversal(ViewRootImpl.java:1000)
- at android.view.ViewRootImpl\$TraversalRunnable.run(ViewRootImpl.java:5670) at android.view.Choreographer\$CallbackRecord.run(Choreographer.java:761)
- at android.view.Choreographer.doCallbacks(Choreographer.java:574)
- at android.view.Choreographer.doFrame (Choreographer.java.57)
- at android.view.Choreographer\$FrameDisplayEventReceiver.run(Choreographer.java:747)
- at android.os.Handler.handleCallback(Handler.java:733)
- at android.os.Handler.dispatchMessage(Handler.java:95)
- at android.os.Looper.loop(Looper.java:136)
- at android.app.ActivityThread.main(ActivityThread.java:5017)
- at java.lang.reflect.Method.invokeNative(Native Method)
- at java.lang.reflect.Method.invoke(Method.java:515)
- at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:779) at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:595)
- at dalvik.system.NativeStart.main(Native Method)

该问题在Android诸多系统版本均有出现。这个问题是由于Android系统的bug导致Webview内存无法被正确管理,从而导致内存泄漏。当内存太低时,绘图时无法申请到 内存资源,导致此异常抛出,应用崩溃。

### 解决方法如下:

1:强制管理Webview的生命周期。在Webview对应的Activity的声明周期管理中,对webview进行强制的管理。当Activity在 onPause() 时,强制destory掉 webview,并清空webview的container。这样,webview占用的内存会在下一次GC时,被系统回收。

@Override public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle saved) { View v = super.onCreateView(inflater, container, saved); mActivity = (BaseActivity) getSupportActivity(); // this is the framelayout which will contain our WebView mWebContainer = (FrameLavout) v.findViewBvId(R.id.web container); return v; public void onResume() { super.onResume(); // create new WebView and set all its options. mWebView = new WebView(mActivity); mWebView.... // add it to the container mWebContainer.addView(mWebView); // if data is available, display it immediately if(mTopic != null) { mWebView.loadDataWithBaseURL("file:///android_asset/", mTopic.getHtmlCache(), "text/html", "UTF-8", null); } ØOverride public void onPause() { super.onPause(); // destroy the webview mWebView.destroy(); mWebView = null; // remove the view from the container. mWebContainer.removeAllViews();

#### 2: 取消应用的硬件加速

在 AndroidManifest.xml 里增加 android:hardwareAccelerated="false",用来禁止硬件加速,此方法可以避免该系统bug。不过,不推荐游戏或者对流畅性要求很高的应用采用此方法,可能会导致应用变得有一些卡顿。

#### 示例三

```
01-17 00:19:44.703: E/Surface(9731): Surface::lock failed, already locked
  01-17 00:19:44.796: E/SurfaceHolder(9731): Exception locking surface
  01-17 00:19:44.796: E/SurfaceHolder(9731): java.lang.IllegalArgumentException
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.Surface.lockCanvasNative(Native Method)
01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.Surface.lockCanvas(Surface.java:314)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.SurfaceView$3.internalLockCanvas(SurfaceView.java:762)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.SurfaceView$3.lockCanvas(SurfaceView.java:741)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at com.frequency.FreqTapArea$2.onTouch(FreqTapArea.java:54)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.View.dispatchTouchEvent(View.java:3897)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.ViewGroup.dispatchTouchEvent(ViewGroup.java:869)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.ViewGroup.dispatchTouchEvent(ViewGroup.java:869)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.ViewGroup.dispatchTouchEvent(ViewGroup.java:869)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at com.android.internal.policy.impl.PhoneWindow$DecorView.superDispatchTouchEvent(PhoneWindow.java:
1737)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at com.android.internal.policy.impl.PhoneWindow.superDispatchTouchEvent(PhoneWindow.java:1153)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.app.Activity.dispatchTouchEvent(Activity.java:2096)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at com.android.internal.policy.impl.PhoneWindow$DecorView.dispatchTouchEvent(PhoneWindow.java:1721)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.ViewRoot.deliverPointerEvent(ViewRoot.java:2200)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.view.ViewRoot.handleMessage(ViewRoot.java:1884)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.os.Handler.dispatchMessage(Handler.java:99)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at android.os.Looper.loop(Looper.java:130)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at android.app.ActivityThread.main(ActivityThread.java:3835)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at java.lang.reflect.Method.invokeNative(Native Method)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at java.lang.reflect.Method.invoke(Method.java:507)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:847)
  01-17 00:19:44.796: E/SurfaceHolder(9731): at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:605)
 01-17 00:19:44.796: E/SurfaceHolder(9731): at dalvik.system.NativeStart.main(Native Method))
```

Canvas在操作之前需要加锁(lockCanvas函数),操作结束后,应该立即释放锁资源(unlockCanvasAndPost函数),这个问题的出现就是因为 lockCanvas() 之后没 有释放锁资源,正确的流程是:

1. 调用 mSurfaceHolder.lockCanvas(); 获取Canvas对象。

2. 调用Canvas方法进行绘图: canvas.drawXXX(arg, arg, arg, arg);

```
3. 绘图完成之后,调用 mSurfaceHolder.unlockCanvasAndPost(c); 释放锁资源。
```

#### 示例如下:

```
public boolean draw(View arg0, MotionEvent arg1) {
    Canvas c = mSurfaceHolder.lockCanvas();
    // Draw something
    c.drawCircle(args,...,arg);
    mSurfaceHolder.unlockCanvasAndPost(c);
    return true;
}
```

# 参考文章

- https://developer.android.com/reference/java/lang/IllegalArgumentException.html
- http://stackoverflow.com/questions/20554621/after-a-while-of-usage-my-app-freezes-during-scrolling-a-webview-saying-could
- http://stackoverflow.com/questions/8875702/android-canvas-locking-throws-illegalargument exception
- http://androidxref.com/4.4_r1/xref/external/apache-http/src/org/apache/http/params/HttpProtocolParams.java#148

# 12.1.16. java_lang_IllegalStateException

#### 问题分析

当一个方法在不合法或不恰当的时间被调用时抛出。

### 解决方案

当一个方法在不合法或不恰当的时间被调用时抛出。建议根据状态图检查对象当前状态,更改错误的调用方法时机。

#### 示例一

- java.lang.IllegalStateException
  - at io.er.yune.MediaPlayer._reset(NativeMethod) at io.er.yune.MediaPlayer.reset(MediaPlayer.java:734)
  - at io.er.yune.widget.VideoView.release(VideoView.java:502)
- at io.er.yune.widget.VideoView.access\$1500(VideoView.java:69)
- at io.er.yune.widget.VideoView\$3.surfaceDestroyed(VideoView.java:164)
- at android.view.SurfaceView.updateWindow(SurfaceView.java:594)
- at android.view.SurfaceView.onWindowVisibilityChanged(SurfaceView.java:239)
- at android.view.View.dispatchWindowVisibilityChanged(View.java:8170)
- at android.view.ViewGroup.dispatchWindowVisibilityChanged(ViewGroup.java:1113)
- at android.view.ViewGroup.dispatchWindowVisibilityChanged(ViewGroup.java:1113) at android.view.ViewGroup.dispatchWindowVisibilityChanged(ViewGroup.java:1113)
- at android.view.ViewGroup.dispatchWindowVisibilityChanged(ViewGroup.java:1113)

### 问题分析:

这种异常是由于MediaPlayer的生命周期使用不正确,比如没有调用prepare等方法就先调用isPlaying等方法时,或非合法状态调用prepareAsyn,setDataSource方法时会 抛出这种异常。

解决方案:由于MediaPlayer的生命周期使用不正确导致抛出的异常,建议阅读MediaPlayer的状态图,确认不同状态调用正确的函数。

#### 代码示例

#### 步骤一:初始化操作

// 方案一:通过将音频文件添加到资源结构中的res/raw文件夹中。

MediaPlayer player =MediaPlayer.create(getApplicationContext(), R.raw.music);

#### // 方案二: 以file://开头的本地文件URI

MediaPlayer player =MediaPlayer.create(getApplicationContext(),Uri.parse("file:///sdcard/music/1.mp3"));

#### // 方案三: set Dat aSource方法

MediaPlayer player =newMediaPlayer();
 player.setDataSource("/sdcard/test.3gp");
 player.prepare();

#### 步骤二:播放控制

关键状态及函数说明:

准备好MediaPlayer后,想要开始播放,必须调用 start() 方法。当此方法成功返回时,MediaPlayer的对象处于Started状态。

isPlaying() 方法可以被调用来测试某个MediaPlayer对象是否在Started状态。而对一个已经处于Started状态的MediaPlayer对象调用 start() 方法没有影响。

 start()
 方法调用之后调用
 pause()
 方法并返回时,播放可以被暂停会使MediaPlayer对象进入Paused状态。调用
 start()
 方法会让一个处于Paused状态的

 MediaPlayer对象从之前暂停的地方恢复播放。当调用
 start()
 方法返回的时候,MediaPlayer对象的状态会又变成Started状态。在Started与Paused状态的相互转换在

 内部的播放引擎中是异步的。所以可能需要一点时间在
 isplaying()
 方法中更新状态,若在播放流内容,这段时间可能会有几秒钟。

调用 stop() 方法会停止播放,并且还会让一个处于Started、Paused、Prepared或PlaybackCompleted状态的MediaPlayer进入Stopped状态。

调用 seekTo() 方法可以调整播放的位置。 seekTo(int) 方法是异步执行的,所以它可以马上返回,但是实际的定位播放操作可能需要一段时间才能完成,尤其是播 放流形式的音频/视频。当实际的定位播放操作完成之后,内部的播放引擎会调用客户端程序员提供的 OnSeekComplete.onSeekComplete() 回调方法。可以通过

setOnSeekCompleteListener(OnSeekCompleteListener) 方法注册。 seekTo(int) 方法也可以在其它状态下调用,比如Prepared、Paused和PlaybackCompleted状态。

播放位置可以调用 getCurrentPosition() 方法得到,它可以帮助如音乐播放器的应用程序不断更新播放进度。

```
private void startPlay() {
    player.start();
}
private void stopPlay() {
    payer.stop();
}
private void pausePlay() {
    player.pause();
}
private void goTo(int pos) {
    player.seekTo(pos);
}
```

#### 步骤三:释放播放资源

当播放到流的末尾,播放就完成了。如果调用了 setLooping (boolean) 方法开启了循环模式,那么这个MediaPlayer对象会重新进入Started状态。

在播放结束时,应调用 MediaPlayer 的 release() 方法,释放播放程序所占用的资源。

```
private void release() {
    if(player != null) {
        player.stop();
        player.release();
        player = null;
    }
}
```

#### 参考文献

- https://developer.android.com/reference/android/media/MediaPlayer.html#StateDiagram (MediaPlayer状态图)
- https://developer.android.com/reference/android/media/MediaPlayer.html (MediaPlayer官方文档)
- http://www.devdiv.com/Android-MediaPlayer%E7%9A%84%E4%BD%BF%E7%94%A8-thread-130166-1-1.html
- http://blog.csdn.net/yearafteryear/article/details/8966221

# 12.1.17. java_lang_NoClassDefFoundError

## 问题分析

应用编译通过但是在运行时本来预期是找得到类文件的但是因为打包等问题无法定位需要加载的类因此在运行期报错。

## 解决方案

NoClassDef FoundError是编译通过,运行时本来预期是找得到类文件的但是因为打包等问题虚拟机无法定位需要加载的类因此在运行期报错。类依赖的class或者jar不存在 请确认打包方式正确。若类文件存在,但是存在不同的域中,此时含有包名的类在编译时需要指定classpath的路径,并且使用的时候指定包名全路径即可。

#### 参考文献

https://docs.oracle.com/javase/7/docs/api/java/lang/NoClassDefFoundError.html

http://blog.csdn.net/jamesjxin/article/details/46606307

# 12.1.18. java_lang_NoSuchMethodException

# 问题分析

找不到特定的函数时抛出此类异常,因为编译时依赖的库版本有这个函数的,但是运行时却没有找到,这个异常通常发生在反射调用时被抛出。

# 解决方案

找不到特定的函数时抛出此类异常,请注意打包的类库版本信息,一般发生在反射调用时被抛出。请查看日志中此Crash附近日志信息,并检查日志报错对应的代码是否 写错函数名。

#### 代码示例

package com.alibaba.mqc.test; import java.lang.reflect.InvocationTargetException; import java.lang.reflect.Method; //@#B0KBidemo public class V { public void printf(){ System.out.println("printf"); } public static void main(String[] args) throws ClassNotFoundException, SecurityException, NoSuchMethodException, IllegalArgumentException, IllegalAccessException, InvocationTargetException, InstantiationException { Class c = Class.forName("com.alibaba.mqc.test.V"); Method m = c.getDeclaredMethod("printf",null); m.invoke(c.newInstance(),null); } }

# 参考文献

https://docs.oracle.com/javase/7/docs/api/java/lang/NoSuchMethodException.html

• http://stackoverflow.com/questions/19913970/java-java-lang-nosuchmethodexception

# 12.1.19. java_lang_NullPointerException

#### 问题分析

试图对一个为null的对象执行成员函数、试图获取null对象的成员变量、试图获取null数组的长度、试图访问数组中某个空对象、抛出一个对象而实际未初始化为null时, 此时会抛出NullPointerException异常。

## 解决方案

对可以预见为空的对象进行处理,对不可预知的对象判断是否是null,然后再访问内部成员变量或执行成员函数。

#### 代码示例

#### 示例1

Exceptionin thread "main" java.lang.NullPointerException

#### 判断对象是否为null,不是null再执行对象成员函数。

```
publicString[] split(String content){
  if(content!=null){
   String[] result=content.split("\\s+");
  return result;
  }
  returnnull;
}
```

# 示例2

java.lang.NullPointerException

- at android.webkit.WebViewClassic\$WebViewInputConnection.setNewText(WebViewClassic.java:587)
- at android.webkit.WebViewClassic\$WebViewInputConnection.setComposingText(WebViewClassic.java:327)
- at android.webkit.WebViewClassic\$WebViewInputConnection.commitText(WebViewClassic.java:343)
- at com.android.internal.view.IInputConnectionWrapper.executeMessage(IInputConnectionWrapper.java:279)
- at com.android.internal.view.IInputConnectionWrapper \$MyHandler.handleMessage (IInputConnectionWrapper.java: 77)
- at android.os.Handler.dispatchMessage(Handler.java:107)
- at android.os.Looper.loop(Looper.java:194)
- at android.app.ActivityThread.main(ActivityThread.java:5391)
- at java.lang.reflect.Method.invokeNative(Native Method)
- at java.lang.reflect.Method.invoke(Method.java:525)
- at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:833)
- at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:600)
- at dalvik.system.NativeStart.main(Native Method)

在使用 Google Play Services的google-play-services.jar 时报错,建议设置全局Exception Handler。

```
public class MyApplication extends Application {
  @Override
  public void onCreate() {
    super.onCreate();
    Thread.setDefaultUncaughtExceptionHandler(new Thread.UncaughtExceptionHandler() {
      @Override
      public void uncaughtException(Thread thread, final Throwable ex) {
          // Custom code here to handle the error.
      }
   });
 }
}
```

# 参考文献

- https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html
- http://stackoverflow.com/questions/21866459/android-nullpointerexception-on-webview
- https://groups.google.com/forum/#!topic/google-admob-ads-sdk/QvUhH5ZGIHQ
- http://stackoverflow.com/questions/18824076/how-to-catch-this-exception-in-android-webview

# 12.1.20. java_lang_NumberFormatException

### 问题分析

通过字符串转数值类型时因为字符串格式不正确无法转换抛出此类异常。

# 解决方案

通过字符串转数值类型时因为字符串格式不正确无法转换抛出此类异常。建议检查转换的字符串并对代码进行捕获。

#### 代码示例

```
public static int parseInt(String s) {
  try {
    return Integer.parseInt(s);
    catch(NumberFormatException e) {
        e.printStackTrace();
    }
}
```

# 参考文献

http://docs.oracle.com/javase/7/docs/api/java/lang/NumberFormatException.html

# 12.1.21. java_lang_OutOfMemoryError

### 问题分析

Out Of MemoryError是在当程序需要请求一块内存,而Java虚拟机进行垃圾回收机制GC后无法再提供要求的内存时抛出此异常。

### 解决方案

OutOf MemoryError是在当程序需要请求一块内存,而Java虚拟机进行垃圾回收机制GC后无法再提供要求的内存时抛出此类异常,程序已经占用的内存到达系统限制的应 用内存上线无法再申请到所需的内存空间。需要重点检查代码中是否有死循环或递归调用。检查是否在循环重复产生占用大内存的新对象。是否有一次获得大量数据的查 询存到内存中,应当尽量用分页的方式查询。是否List、Map等集合对象使用后未清除,而集合对象有对对象的引用导致对象不能被GC回收,是否有内存泄露等。建议使 用深度性能测试检测内存泄漏等问题。

### 代码示例

```
// 在循环外初始化大对象list并在循环内重复使用,并清空集合类、置null
List list=new ArrayList<Object>();
for(Object tmp:otherList){
    // to do something
}
list.clear();
list=null;
```

### 参考文献

- https://developer.android.com/reference/java/lang/OutOfMemoryError.html
- https://source.android.com/compatibility/android-cdd.html#7_6_memory_and_storage (安卓系统对程序所使用内存的最大值限制,参见3.7节的表格,或直接参考文献3)
- https://drive.google.com/file/d/0B7Vx10vzrLa3Y0R0X1BZbUpicGc/view?usp=sharing (安卓系统对内存最大值限制)

# 12.1.22. java_lang_reflect_InvocationTargetException

### 问题分析

InvocationTargetException异常由 Method.invoke(obj, args...) 方法抛出。当被调用的方法的内部抛出了异常而没有被捕获时,由此异常接收。

# 解决方案

在反射调用方法时,被调用的方法内部抛出了异常没有被捕获会抛出此类异常,建议使用try Catch块进行捕获并处理。

### 代码示例

```
public static void main(String[] args) {
   try {
      Class<?> clazz = Class.forName("com.alibaba.test");
      Method method = clazz.getMethod("run", int.class);
      method.invoke(clazz.newInstance(), -1);
   } catch (InvocationTargetException e) {
      System.out.println("此处接收被调用方法内部未被捕获的异常");
      e.printStackTrace();
   }
}
```

### 详细的例子:

}

java.lang.reflect.InvocationTargetException

- at java.lang.reflect.Method.invokeNative(Native Method)
- at java.lang.reflect.Method.invoke(Method.java:515)
- at android.app.ActivityThread.pauseGC(ActivityThread.java:5525)
- at and roid.app.ActivityThread.performLaunchActivity(ActivityThread.java:2324)
- at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2471)
- at android.app.ActivityThread.access\$900(ActivityThread.java:175)
- at android.app.ActivityThread\$H.handleMessage(ActivityThread.java:1308) at android.os.Handler.dispatchMessage(Handler.java:102)
- at android.os.Looper.loop(Looper.java:146)
- at android.app.ActivityThread.main(ActivityThread.java:5602)
- at java.lang.reflect.Method.invokeNative(Native Method)
- at java.lang.reflect.Method.invoke(Method.java:515)
- at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit.java:1283)
- at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1099)
- at dalvik.system.NativeStart.main(Native Method)
- Caused by: java.lang.UnsatisfiedLinkError: Native method not found: dalvik.system.VMRuntime.pauseGc:(Ljava/lang/String;)I

at dalvik.system.VMRuntime.pauseGc(Native Method) ... 15 more

针对这个异常的堆栈信息来看,暂时没有好的解决方案,这个问题是三星Android 4.4.2、4.4.4版本引入的bug,原因是:在ActivityThread文件的第5525行调用反射调用 native方法 pauseGc() 而未找到此函数。每当视频重绘调用GLSurfaceView时,会起一个新的Activity会产生2MB的垃圾,但GC机制没有对此进行垃圾回收。因此如果播放一个视频30min左右内存就满了,应用Crash产生。

#### 参考文献

- http://developer.samsung.com/forum/thread/pause-gc-error-post-android-442-upgrade-on-samsung-galaxy-note-ii/201/266242? boardName=SDK&startId=zzzzz~&searchSubId=000000029
- https://docs.oracle.com/javase/7/docs/api/java/lang/reflect/InvocationTargetException.html

# 12.1.23. java_lang_RuntimeException

#### 问题分析

RuntimeException是运行时异常,是Java编译器事先不可预见的异常。RuntimeException的子类众多,这里列举一些比较常见的子类:

- BufferOverflowException当相关put操作达到目标缓冲区限制时, 抛出此未经检查的异常。
- ArrayStoreException试图将错误类型的对象存储到一个对象数组时抛出的异常。
- ArithmeticException当出现异常的运算条件时,抛出此异常。例如,一个整数"除以零"时,抛出此类的一个实例。
- BufferUnderflowException当相关get操作达到源缓冲区限制时,抛出此未经检查的异常。
- IndexOutOfBoundsException指示某排序索引(例如对数组、字符串或向量的排序)引用时超出范围时抛出。
- NoSuchElement Exception由Enumeration的nextElement方法抛出,表明枚举中没有更多的元素。

#### 解决方案

RuntimeException是发生在程序运行期,预先不可预见的发生。编译器未要求一定要进行捕获,如果运行期没有处理,则RuntimeException会一直往上层抛。最后由IVM 来处理,JVM会打印堆栈信息然后结束应用。对于可能发生的RuntimeException,建议根据堆栈信息,检查代码是否有误并进行更改,如果情况复杂无法全部解决,可以 对RuntimeException进行捕获并进行业务恢复处理。

#### 代码示例

```
try {
   somethingThrowingARuntimeException();
   }catch (RuntimeException e) {
        // Do something with it. At least log it
        e.printStackTrace();
}
```

#### 示例一

java.lang.RuntimeException:Unable to start activity ComponentInfo{*}: android.support.v4.app.Fragment\$InstantiationException:*: make sure class n ame exists, is public, and has an empty constructor that is public

当系统因为内存不足杀死非前台进程,用户又将被系统杀掉的非前台应用带回前台,如果这个时候有U是呈现在Fragment中,那么会因为rest ore造成fragment需要通过 反射实例对象,从而将之前save的状态还原。因此不应该改写构造函数,而应该使用静态方法比如newInstance来构造有参Fragment对象并返回。

#### 代码示例:

}

### 步骤一:编写有参静态创建对象函数

```
public static final AlertFragment newInstance(int title, String message)
{
    AlertFragment f = new AlertFragment();
    Bundle bdl = new Bundle(2);
    bdl.putInt(EXTRA_TITLE, title);
    bdl.putString(EXTRA_MESSAGE, message);
    f.setArguments(bdl);
    return f;
```

#### 步骤二:通过Bundle获取参数

```
@Override
public void onCreate(Bundle savedInstanceState)
{
    title = getArguments().getInt(EXTRA_TITLE);
    message = getArguments().getString(EXTRA_MESSAGE);
    //...
    //etc
    //...
}
```

#### 步骤三:通过Fragment Manager对Fragment进行管理。

#### 示例二

```
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 80, Size: 3
```

- at java.util.ArrayList.rangeCheck(ArrayList.java:635)
- at java.util.ArrayList.remove(ArrayList.java:474)
- at com.alibaba.mqc.test.Test.indexOutOfBounds(Test.java:66) at com.alibaba.mqc.test.Test.main(Test.java:32)

# 问题分析:

IndexOut Of BoundsException 指示某排序索引(例如对数组、字符串或向量的排序)引用时超出范围时抛出。请检查是否有对List有越界索引,判断访问数组的索引是否 合法。

# 有误的代码:

```
public void indexOutOfBounds() {
    ArrayList<Character> a = new ArrayList<Character>();
    a.add('A');
    a.add('C');
    a.add('C');
    System.out.println(a);
    //a.remove('A');
    a.remove(80); // 抛出此类异常
    System.out.println(a);
}
```

#### 建议的代码:

```
public void indexOutOfBounds() {
    ArrayList<Character> a = new ArrayList<Character>();
    a.add('A');
    a.add('B');
    a.add('C');
    System.out.println(a);
    if(a.size()>80) {
        a.remove(80);
    }
    System.out.println(a);
}
```

# 参考文献

https://developer.android.com/reference/java/lang/RuntimeException.html

- http://stackoverflow.com/questions/10450348/do-fragments-really-need-an-empty-constructor
- http://blog.csdn.net/xplee0576/article/details/43057633
- http://stackoverflow.com/questions/2028719/handling-runtimeexceptions-in-java
- http://tool.oschina.net/uploads/apidocs/jdk-zh/java/lang/RuntimeException.html

# 12.1.24. java_lang_SecurityException

#### 问题分析

当违背安全原则,检测到安全错误时会抛出此类异常。

### 解决方案

当违背安全原则,检测到安全错误时会抛出此类异常。具体原因需要看给出的相应的其他信息并获取相应权限。

#### 示例

- Caused by: java.lang.SecurityException: Permission denied (missing INTERNET permission?)
  - at java.net.InetAddress.lookupHostByName(InetAddress.java:430)
  - at java.net.InetAddress.getAllByNameImpl(InetAddress.java:236) at java.net.InetAddress.getAllByName(InetAddress.java:214)
  - at libcore.net.http.HttpConnection.<init>(HttpConnection.java:70)
  - at libcore.net.http.HttpConnection.<init>(HttpConnection.java:50)
  - at libcore.net.http.HttpConnection\$Address.connect(HttpConnection.java:340)
  - at libcore.net.http.HttpConnectionPool.get(HttpConnectionPool.java:87)
  - at libcore.net.http.HttpConnection.connect(HttpConnection.java:128)
  - at libcore.net.http.HttpEngine.openSocketConnection(HttpEngine.java:316)

#### 问题分析:

因为没有获取到访问网络权限,因此抛出此类异常。

#### 解决方案:

因为没有获取到访问网络权限而抛出此类异常。需要检查 AndroidManifest.xml 文件是否获取权限,检查手机是否有限制权限的软件如管家等并打开权限。

#### 代码示例

在 AndroidManifest.xml 中使用网络权限。

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

### 因为在这个例子中使用HTTPUrlConnection类,根据官方文档还需要申请检测网络状态的权限。

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

#### 如果还是报这个异常,请确认手机中没有其他软件管家控制网络权限没有授权。

#### 参考文献

- https://developer.android.com/reference/java/lang/SecurityException.html
- https://developer.android.com/training/basics/network-ops/connecting.html

# 12.1.25. java_lang_UnsatisfiedLinkError

### 问题分析

JVM找不到对应的native方法抛出此类异常。

#### 解决方案

JVM找不到对应的native方法抛出此类异常,请检查.so文件是否存在、位置是否正确以及是否加载成功。通常是因为应用运行于不同的手机之上,CPU架构不同,因此需 要编译生成支持不同CPU架构的静态库so文件。

java.lang.UnsatisfiedLinkError:Native method not found: com.baidu.android.moplus.systemmonitor.security.md5.MD5.Transform_native:([[[BII)V

- at com.c.h.systemmonitor.security.md5.MD5.Transform native(NativeMethod)
- at com.c.h.systemmonitor.security.md5.MD5.update(UnknownSource)
- at com.c.h.systemmonitor.security.md5.MD5.update(UnknownSource)
- at com.c.h.systemmonitor.security.md5.MD5InputStream.read(UnknownSource)
- at java.io.InputStream.read(InputStream.java:163)

### 代码示例

步骤一:编写 Application.mk 文件

在jni目录下(即hello-jni.c同级目录下)新建一个 Application.mk 文件,在文件中添加如下代码:

APP_ABI := armeabi armeabi-v7a x86

这段话表示你可以同时生成三个处理器的so库。如果没有或不想使用 Application.mk 文件,则在ndk-build参数中添加:

APP_ABI="armeabi armeabi-v7a x86 mips"

#### 即运行:

ndk-build APP_ABI="armeabi armeabi-v7a x86 mips"

#### 步骤二: 生成.so共享库文件

Android.mk 文件已经编写好了,现在可以用Android NDK开发包中的ndk-build脚本生成对应的.so共享库了,方法如下:

mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ ls
AndroidManifest.xml assets bin default.properties gen jni libs obj res src
mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ ndk-build
Gdbserver : [arm-linux-androideabi-4.4.3] libs/armeabi/gdbserver
Gdbsetup : libs/armeabi/gdb.setup
Install : libhello-jni.so => libs/armeabi/libhello-jni.so

这样,在文件夹 hello-jni/libs 下生成了三个文件夹armeabi/armeabi-v7a/x86,里面已经正确的生成了 libhello-jni.so 共享库了。

#### 步骤三:在eclipse重新编译HelloJni工程,生成apk

eclipse中刷新下HelloJni工程,重新编译生成apk,libhello-jni.so共享库会一起打包在apk文件内。

#### 附录

NDK程序开发详解:

#### 步骤一:搭建开发环境

1: Android的NDK开发需要在Linux下进行:因为需要把C/C++编写的代码生成能在arm上运行的.so文件,这就需要用到交叉编译环境,而交叉编译需要在Linux系统下才 能完成。

- 2: 安装android-ndk开发包,这个开发包可以在Google Android官网下载,通过这个开发包的工具才能将android jni的C/C++的代码编译成库。
- 3: Android应用程序开发环境:包括 eclipse、java、android sdk、adt 等。
- 4:安装完成android-ndk之后,将android-ndk的路劲加到环境变量PATH中:

sudo gedit /etc/environment

```
5.在environment的PATH环境变量中添加android-ndk的安装路径,然后再让这个更改的环境变量立即生效: source /etc/environment
```

#### 经过了上述步骤,在命令行下输入:

ndk-bulid

#### 弹出如下的错误,而不是说ndk-build not found,就说明ndk环境已经安装成功了。

Android NDK: Could not find application project directory ! Android NDK: Please define the NDK_PROJECT_PATH variable to point to it. /home/mqc/workspace/android/android-ndk-r5/build/core/build-local.mk:85: *** Android NDK: Aborting . Stop.

#### 步骤二:编写 Java 代码

#### 建立一个Android应用工程HelloJni, 创建 HelloJni.java 文件。

```
import android.app.Activity;
import android.widget.TextView;
import android.os.Bundle;
public class HelloJni extends Activity
    /** Called when the activity is first created. */
   ROverride
   public void onCreate (Bundle savedInstanceState)
        super.onCreate(savedInstanceState);
       TextView tv = new TextView(this);
       tv.setText( stringFromJNI() );
        setContentView(tv);
   /* A native method that is implemented by the 'hello-jni' native library, which is packaged with this application. */
   public native String stringFromJNI();
   public native String unimplementedStringFromJNI();
   /* this is used to load the 'hello-jni' library on application startup. The library has already been unpacked into
     /data/data/com.example.HelloJni/lib/lib/ello-jni.so at installation time by the package manager. */
    static {
       System.loadLibrary("hello-jni");
}
```

上面展示的代码中的static代码块表明程序开始运行的时候会加载hello-jni, static区声明的代码会先于onCreate方法执行。如果你的程序中有多个类,而且如果HelloJni这 个类不是你应用程序的入口,那么 hello-jni(完整的名字是 libhello-jni.so )这个库会在第一次使用HelloJni这个类的时候加载。

publicnativeString stringFromJNI();
publicnativeString unimplementedStringFromJNI();

上面展示的代码中可以看到这两个方法的声明中有native关键字,这个关键字表示这两个方法是本地方法,也就是说这两个方法是通过本地代码(C/C++)实现的,在 Java代码中仅仅是声明。使用eclipse编译该工程,生成相应的.class文件,因为生成.h文件需要用到相应的.class文件。

### 步骤三:编写相应的 C/C++ 代码

利用javah生成相应的.h文件,然后根据这个.h文件编写相应的C/C++代码。

1:生成相应.h文件。

首先在终端下进入刚刚建立的 HelloJni 工程的目录:

mqc@ubuntu:~\$ cd workspace/android/NDK/hello-jni/

#### 查看工程文件。

mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ ls AndroidManifest.xml assets bin default.properties gen res src

可以看到目前仅仅有几个标准的Android应用程序的文件(夹)。首先我们在工程目录下建立一个jni文件夹:

mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ mkdir jni mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ ls AndroidManifest.xml assets bin default.properties gen jni res src

#### 下面就可以生成相应的.h文件了:

mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ javah -classpath bin -d jni com.example.hellojni.HelloJni

- -classpath bin :表示类的路劲
- -d jni : 表示生成的头文件存放的目录
- com.example.hellojni.HelloJni 则是完整类名

这一步的成功要建立在已经在 bin/com/example/hellojni/ 目录下生成了 HelloJni.class 的基础之上。现在可以看到jni目录下多了个.h文件:

mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ cd jni/ mqc@ubuntu:~/workspace/android/NDK/hello-jni/jni\$ ls com_example_hellojni_HelloJni.h

下面展示出 com_example_hellojni_HelloJni.h 的内容:

#### com_example_hellojni_HelloJni.h:

```
/* DO NOT EDIT THIS FILE - it is machine generated */
  #include <jni.h>
  /* Header for class com_example_hellojni_HelloJni */
  #ifndef _Included_com_example_hellojni_HelloJni
  #define _Included_com_example_hellojni_HelloJni
  #ifdef __cplusplus
extern "C" {
  #endif
  /*
  / * Class: com_example_hellojni_HelloJni
 * Method: stringFromJNI
   * Signature: ()Ljava/lang/String;
   * /
  JNIEXPORT jstring JNICALL Java_com_example_hellojni_HelloJni_stringFromJNI
    (JNIEnv *, jobject);
  /*
  * Class: com_example_hellojni_HelloJni
* Method: unimplementedStringFromJNI
   * Signature: ()Ljava/lang/String;
   * /
  JNIEXPORT jstring JNICALL Java_com_example_hellojni_HelloJni_unimplementedStringFromJNI
    (JNIEnv *, jobject);
  #ifdef __cplusplus
  #endif
  #endif
```

### 上面的源码中函数名是很有规律的,完全按照: java_pacakege_class_mathod形式来命名,即:

Hello.java 中 stringFromJNI() 方法对应于 C/C++中的 Java_com_example_hellojni_HelloJni_stringFromJNI() 方法。

• HelloJni.java 中的 unimplementedStringFromJNI() 方法对应于 C/C++中的 Java_com_example_helloJni_HelloJni_unimplementedStringFromJNI() 方法。

#### 步骤四:编写相应的.c文件

#### hello-jni.c :

<pre>#include <string.h> #include <jni.h> /* This is a trivial JNI example where we use a native method * to return a new VM String. See the corresponding Java source</jni.h></string.h></pre>
<pre>* file located at: * apps/samples/hello-jni/project/src/com/example/HelloJni/HelloJni.java */</pre>
<pre>jstring Java_com_example_hellojni_HelloJni_stringFromJNI( JNIEnv* env, jobject thiz ) {     return (*env)-&gt;NewStringUTF(env, "Hello from JNI !"); }</pre>

Java_com_example_hellojni_HelloJni_stringFromJNI() 函数只是简单的返回了一个内容为 "Hello from JNI!" 的jstring对象(对应于Java中的String对象)。

#### 步骤五:编译 hello-jni.c 生成相应的库

# 1: 编写Android.mk文件

在jni目录下(即 hello-jni.c 同级目录下)新建一个 Android.mk 文件, Android.mk 文件是Android的makefile文件, 内容如下:

# Copyright (C) 2009 The Android Open Source Project

# Licensed under the Apache License, Version 2.0 (the "License");

- # you may not use this file except in compliance with the License.
- # You may obtain a copy of the License at
- # http://www.apache.org/licenses/LICENSE-2.0
- #
- $\ensuremath{\texttt{\#}}$  Unless required by applicable law or agreed to in writing, software
- $\ensuremath{\texttt{\#}}$  distributed under the License is distributed on an "AS IS" BASIS,
- # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
- # See the License for the specific language governing permissions and # limitations under the License.
- # 1101104010

*
UCCAL_PATH := \$(call my-dir)
include \$(CLEAR_VARS)
LOCAL_MODULE := hello-jni
LOCAL_SRC_FILES := hello-jni.c
include \$(BUILD_SHARED_LIBRARY)

#### 这个 Androd.mk 文件很短,下面我们来逐行解释下:

LOCAL_PATH := \$(call my-dir)

```
一个 Android.mk 文件首先必须定义好 LOCAL_PATH 变量。它用于在开发树中查找源文件。在这个例子中,宏函数 my-dir ,由编译系统提供,用于返回当前路径
(即包含 Android.mk file文件的目录)。
```

include \$( CLEAR_VARS)

CLEAR_VARS由编译系统提供,指定让GNU MAKEFILE为你清除许多LOCAL_XXX变量(例如LOCAL_MODULE、LOCAL_SRC_FILES、LOCAL_STATIC_LIBRARIES等),除 LOCAL_PATH。这是必要的,因为所有的编译控制文件都在同一个GNU MAKE执行环境中,所有的变量都是全局的。

LOCAL_MODULE := hello-jni

编译的目标对象,LOCAL_MODULE变量必须定义,以标识你在 Android.mk 文件中描述的每个模块。名称必须是唯一的,而且不包含任何空格。

#### ↓ 注意

- 编译系统会自动产生合适的前缀和后缀,换句话说,一个被命名为hello-jni的共享库模块,将会生成 libhello-jni.so 文件。
- 如果你把库命名为lbhello-jni,编译系统将不会添加任何的lb前缀,也会生成 libhello-jni.so ,这是为了支持来源于Android平台的源代码的 Android.mk 文件。

LOCAL_SRC_FILES := hello-jni.c

LOCAL_SRC_FILES变量必须包含将要编译打包进模块中的C或C++源代码文件。

#### ↓ 注意

你不用在这里列出头文件和包含文件,因为编译系统将会自动为你找出依赖型的文件;仅仅列出直接传递给编译器的源代码文件就好。

include \$(BUILD_SHARED_LIBRARY)

 BUILD_SHARED_LIBRARY表示编译生成共享库,是编译系统提供的变量,指向一个GNU Makefile脚本,负责收集自从上次调用 include \$(CLEAR_VARS)
 UR,定义在

 LOCAL_XXX变量中的所有信息,并且决定编译什么,如何正确地去做。还有BUILD_STATIC_LIBRARY变量表示生成静态库:
 11b\$(LOCAL_MODULE).a

, BUILD_EXECUT ABLE表示生成可执行文件。

2: 编写 Application.mk 文件

在jni目录下(即 hello-jni.c 同级目录下)新建一个 Application.mk 文件,在文件中添加如下代码:

APP_ABI := armeabi armeabi-v7a x86

这段话表示你可以同时生成三个处理器的so库。如果没有或不想使用 Application.mk 文件,则在ndk-build参数中添加:

APP_ABI="armeabi armeabi-v7a x86 mips"

#### 即运行:

ndk-build APP_ABI="armeabi armeabi-v7a x86 mips"

#### 3: 生成.so共享库文件

Android.mk 文件已经编写好了,现在可以用Android NDK开发包中的ndk-build脚本生成对应的.so共享库了,方法如下:

```
mqc@ubuntu:~/workspace/android/NDK/hello-jni$ ls
AndroidManifest.xml assets bin default.properties gen jni libs obj res src
mqc@ubuntu:~/workspace/android/NDK/hello-jni$ ndk-build
Gdbserver : [arm-linux-androideabi-4.4.3] libs/armeabi/gdbserver
Gdbsetup : libs/armeabi/gdb.setup
Install : libhello-jni.so => libs/armeabi/libhello-jni.so
```

#### 可以看到已经正确的生成了 libhello-jni.so 共享库了。

#### 4:在eclipse重新编译HelloJni工程,生成apk

eclipse中刷新下HelloJni工程,重新编译生成apk, libhello-jni.so 共享库会一起打包在apk文件内。

#### 参考文献

- https://developer.android.com/training/articles/perf-jni.html?hl=ja#faq_ULE
- https://www.ibm.com/developerworks/community/blogs/738b7897-cd38-4f24-9f05-48dd69116837/entry/debugging_java_lang_unsatisfiedlinkerror9? lang=en
- http://wiki.jikexueyuan.com/project/jni-ndk-developer-guide/ndk.html (NDK开发教程)
- http://www.cnblogs.com/mengshu-lbq/archive/2013/03/29/2988657.html
- https://my.oschina.net/lhjtianji/blog/188074

# 12.1.26. java_lang_UnsupportedOperationException

# 问题分析

当请求了不支持的操作时抛出此类异常。这个异常有三个子类:

- 1. HeadlessException,在不支持键盘、显示器或鼠标的环境中调用与键盘、显示器或鼠标有关的代码时,被抛出的异常。
- 2. ReadOnlyBufferException,当在只读缓冲区上调用内容可变(content-mutation)的方法(例如put或compact方法)时,抛出此未经检查的异常。

3. ReadOnlyFileSystemException,当试图更新一个关联了只读权限的文件的对象时抛出此类异常。

# 解决方案

当请求了不支持的操作时抛出此类异常,请根据堆栈信息检查相关代码,根据官方文档及状态图检查是否进行了不支持的操作。

### 示例一

```
java.lang.UnsupportedOperationException
    java.util.AbstractList.add(UnknownSource)
    java.util.AbstractList.add(UnknownSource)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:641)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:722)
```

问题分析:根据官方文档, asList() 函数返回的List不支持结构调整,即不支持remove和add操作。

解决方案:可以创建一个新的List,将符合要求的List加入新的List对象中。

#### 代码示例

```
public void chooseObject(String[] array, String key){
   List<String> list=Arrays.asList(array);
   /* 错误的代码
   list.remove(0); */
   List<String> newList=new ArrayList<String>();
   for(String tmp:list) {
      if(tmp.contains(key)) {
         newList.add(tmp);
      }
   }
   System.out.println(newList.toString());
}
```

# 参考文献

- https://docs.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html
- http://docs.oracle.com/javase/6/docs/api/java/util/Arrays.html#asList%28T...%29 (asList函数说明)
- https://stackoverflow.com/questions/5755477/java-list-add-unsupportedoperationexception

# 12.1.27. java_net_MalformedURLException

# 问题分析

在URL格式不正确时抛出此类异常,一般是协议不合法或者字符串无法被解析。

# 解决方案

在URL格式不正确时抛出此类异常,一般是协议不合法或者字符串无法被解析,建议捕获此类异常并做业务处理。

# 代码示例

```
public void parse(String location){
   try
   {
     projectUrl = new URL(location);
   }
   catch (MalformedURLException e)
   {
     e.printStackTrace();
   }
}
```

### 参考文献

https://docs.oracle.com/javase/7/docs/api/java/net/MalformedURLException.html

# 12.1.28. java_net_SocketException

# 问题分析

当服务端创建SocketServer时、客户端向服务端创建socket时、连接过程中发生错误时会抛出此类异常。此类异常主要有四种子类BindException、 ConnectException、 NoRouteToHostException、 PortUnreachableException,均可以被SocketException捕获。这四种子类的问题分析如下:

- 1. BindException试图将套接字绑定到本地地址和端口时发生错误的情况下, 抛出此异常。
- 2. ConnectException试图将套接字连接到远程地址和端口时发生错误的情况下,抛出此异常,典型的方式是因为远程主机拒绝,例如远程主机不存在或并未监听此端口。
- 3. NoRouteToHostException试图将套接字连接到远程地址和端口时发生错误的情况下,抛出此异常,一般原因是因为防火墙拦截或必经路由器中断引起的。
- 4. Port UnreachableException在连接的数据报上已接收到ICMP Port Unreachable消息时, 抛出该异常。

### 解决方案

当服务端创建SocketServer时、客户端向服务端创建socket时、连接过程中发生错误时会抛出此类异常,建议使用try catch finally块捕获异常并做好业务处理。

# 示例一

java.net.SocketException:Connection reset

- at java.net.SocketInputStream.read(SocketInputStream.java:196)
- at java.net.SocketInputStream.read(SocketInputStream.java:122) at sun.nio.cs.StreamDecoder.readBytes(StreamDecoder.java:283)
- at sun.nio.cs.StreamDecoder.implRead(StreamDecoder.java:205
- at sun.nio.cs.StreamDecoder.read(StreamDecoder.java:177)
- at java.io.InputStreamReader.read(InputStreamReader.java:184)
- at java.io.BufferedReader.fill(BufferedReader.java:154)
- at java.io.BufferedReader.readLine(BufferedReader.java:317)
- at java.io.BufferedReader.readLine(BufferedReader.java:382)

问题分析:该异常在客户端和服务器端均有可能发生,引起该异常的原因有两个,第一个就是如果一端的Socket被关闭(或主动关闭或者因为异常退出而引起的关闭), 另一端仍发送数据,发送的第一个数据包引发该异常(Connect reset by peer)。另一个是一端退出,但退出时并未关闭该连接,另一端如果在从连接中读数据则抛出该 异常(Connection reset)。主要原因是在连接断开后的读和写操作引起的异常。

解决方案:建议不要频繁建立过多socket请求导致服务器压力增大抛出SocketException,并使用try catch块捕获异常做后续处理。如果服务是幂等的,则客户端可以重 试连接。如果不是幂等的,重试可能造成重复提单。

代码示例:

#### 1: 服务端代码

```
class SimpleServer implements Runnable {
    @Override
    public void run() {
        ServerSocket serverSocket = null;
        try {
           serverSocket = new ServerSocket(8111);
            serverSocket.setSoTimeout(3000);
            while (true) {
               try {
                  Socket clientSocket = serverSocket.accept();
                   BufferedReader inputReader = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
                   System.out.println("Client said :"+ inputReader.readLine());
               } catch (SocketTimeoutException e) {
                   e.printStackTrace();
               }
           }
        }catch (SocketException e) {
           e.printStackTrace();
        }catch (IOException e1) {
            el.printStackTrace();
        } finally {
           try {
               if (serverSocket != null) {
                   serverSocket.close();
               }
           } catch (IOException e) {
               e.printStackTrace();
            }
       }
   }
}
```

### 2:客户端代码

```
class SimpleClient implements Runnable {
    QOverride
    public void run() {
       Socket socket = null;
        try {
           socket = new Socket("localhost", 8111);
            PrintWriter outWriter = new PrintWriter(socket.getOutputStream(), true);
            System.out.println("Wait");
           Thread.sleep(15000);
           //throw new Exception("Random exception");
           outWriter.println("Hello Mr. Server!");
       }catch (SocketException e) {
           e.printStackTrace();
        }catch (InterruptedException e) {
           e.printStackTrace();
        } catch (UnknownHostException e) {
            e.printStackTrace();
       } catch (IOException e) {
           e.printStackTrace();
       } catch (Exception e) {
           e.printStackTrace();
        } finally {
           try {
               if (socket != null)
                    socket.close();
           } catch (IOException e) {
               e.printStackTrace();
            1
       }
```

# 示例二

- java.net.ConnectException:Connection refused: connect
- at java.net.DualStackPlainSocketImpl.connect0(NativeMethod) at java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:79)
- at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:339)
- at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:200)
- at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:182)
- at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:172)
- at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
- at java.net.Socket.connect(Socket.java:579)
- at java.net.Socket.connect(Socket.java:528) at java.net.Socket.(Socket.java:425)
- at java.net.Socket.(Socket.java:425) at java.net.Socket.(Socket.java:208)
- 问题分析:服务端不能响应客户端的连接请求抛出异常。可能原因:IP或者端口写错、服务器宕机、防火墙等原因。

解决方案:服务端不能响应客户端的连接请求抛出异常。建议检查IP或者端口正确、服务器可以访问、防火墙未限制。代码示例与示例一相同。

# 参考文献

- https://docs.oracle.com/javase/7/docs/api/java/net/SocketException.html
- https://docs.oracle.com/javase/7/docs/api/java/net/BindException.html
- https://docs.oracle.com/javase/7/docs/api/java/net/ConnectException.html
- https://docs.oracle.com/javase/7/docs/api/java/net/NoRouteToHostException.html
- https://examples.javacodegeeks.com/core-java/net/connectexception/java-net-connectexception-how-to-solve-connect-exception/
- http://www.seotcs.com/blog/796.html
- https://my.oschina.net/xionghui/blog/508758

# 12.1.29. java_net_SocketTimeoutException

# 问题分析

这类异常发生在建立socket连接或读取数据时发生超时时抛出。

# 解决方案

SocketTimeoutException发生在socket连接过程中的 accept() 和 read() 中,服务端或客户端设置超时连接后,当到达超时时间均会抛出此类异常。建议设置可接 受的socket超时时间并捕获处理SocketTimeoutException。

# 代码示例

### 服务器端:

```
ServerSocket serverSocket = null;
    try {
       serverSocket = new ServerSocket(8111);
       serverSocket.setSoTimeout(10000);
       while (true) {
           try {
               Socket clientSocket = serverSocket.accept();
               BufferedReader inputReader = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
               System.out.println("Client said :"+ inputReader.readLine());
           } catch (SocketTimeoutException e) {
               e.printStackTrace();
           }
       }
   } catch (IOException el) {
       el.printStackTrace();
    } finally {
       try {
           if (serverSocket != null) {
               serverSocket.close();
           3
       } catch (IOException e) {
           e.printStackTrace();
       }
```

#### 客户端:

```
@Override
public void run() {
   try {
       HttpURLConnection.setFollowRedirects(false);
       HttpURLConnection con = (HttpURLConnection) new URL(url).openConnection();
       con.setRequestMethod("HEAD");
        con.setConnectTimeout(5000);
       log.info("Response Code: " + con.getResponseCode());
    } catch (SocketTimeoutException e) {
       System.out.println("SocketTimeoutException"+e.getMessage());
    } catch (IOException e) {
       System.out.println("IOException"+e.getMessage());
    } catch (Exception e) {
       System.out.println("Exception"+e.getMessage());
    }
}
```

# 参考文献

- https://developer.android.com/reference/java/net/SocketTimeoutException.html
- https://examples.javacodegeeks.com/core-java/net/sockettimeoutexception/java-net-sockettimeoutexception-how-to-solve-sockettimeoutexception/

# 12.1.30. org_json_JSONException

### 问题分析

通过调用 interrupt() 方法,线程在sleep/wait/join时,会抛出InterruptedException,可以在适当地方配合 isInterrupted() 方法检查中断状态,并对异常进行捕获并处理,这样线程不用继续等待,线程控制权直接会转给catch住异常块的代码。

# 解决方案

线程在sleep/wait/join时如果对线程调用interrupt方法,会抛出InterruptedException,对该类异常进行捕获,可以获得线程控制权,继续执行任务。

# 代码示例

```
public class PlayerMatcher {
    private PlayerSource players;
    public PlayerMatcher(PlayerSource players) {
       this.players = players;
    public void matchPlayers() throws InterruptedException {
       try {
             Player playerOne, playerTwo;
             while (true) {
               playerOne = playerTwo = null;
                 // Wait for two players to arrive and start a new game
                playerOne = players.waitForPlayer(); // could throw IE
                playerTwo = players.waitForPlayer(); // could throw IE
                startNewGame(playerOne, playerTwo);
            }
         catch (InterruptedException e) {
             // If we got one player and were interrupted, put that player back
            if (playerOne != null)
                players.addFirst(playerOne);
             // Then propagate the exception
            throw e;
        }
   }
}
```

# 参考文献

- https://docs.oracle.com/javase/7/docs/api/java/lang/InterruptedException.html
- http://www.ibm.com/developerworks/cn/java/j-jtp05236.html
- http://blog.csdn.net/derekjiang/article/details/4845757

# 12.2. 安装时错误

# 12.2.1. FAILED_OLDER_SDK

因为APK编译是在高版本SDK下编译生成的,但是手机设备是低版本。

具体可以查看 AndroidManifest.xml 中 android:minSdkVersion 的值。

而且,随着安卓版本的不断更新,新的API不断提供,有时候高版本的API会在低版本发生crash。

如果 minSdkVersion 设置过低,在build的时候,会报错 (Call requires API level (current min is ))。

为了编译可以通过,可以添加 @SuppressLint("NewApi") 或者 @TargeApi(SAPI_LEVEL) 注解,但这样只是编译通过,运行时依然会Crash。

若要兼容低版本,需要判断运行时的版本,在低版本系统中不调用此方法,同时为了保证功能的完整性,提供低版本功能实现。SDK版本(一个自然数字)可以通过下面 的代码确定:

```
int currentapiVersion = android.os.Build.VERSION.SDK_INT;
if (currentapiVersion >= android.os.Build.VERSION_CODES.FROYO) {
    // 使用高版本的API
} else {
    // 使用低版本的API
```

# 12.2.2. INSTALL_FAILED_CANCELLED_BY_USER

# 问题分析

部分机型在使用adb命令安装应用时,设备会弹窗询问用户是否允许安装,假如一段时间内(一般是30s),如果没有点击确认弹窗此类设备会阻止安装,并提示用户: INSTALL_FAILED_CANCELLED_BY_USER。

#### 解决方案

此时需要单击确认安装的按钮。如果报此类异常,请联系技术支持。

# 12.2.3. INSTALL_FAILED_CONTAINER_ERROR

## 日志示例

D/VoldConnector(553): RMV <- {200 6 asec operation succeeded} E/MountService(553): External SD not exist, and PMS want to create ASEC in SD (APP2SD). For SWAP feature, make createSecureContainer() fail! E/DefContainer(7417): Failed to create container smdl2tmplI/ProviderMap(553): getProviderByName: , callingUid = 1000 W/ActivityManager(553): No content provider found for permission revoke: file:///data/local/tmp/a.apk I/PackageManager(553): Apk copy done

#### 问题分析

Mount Service要求读写外部存储器权限,但是存储器不存在,因此安装失败返回。

#### 解决方案

KeySolution:需要安装SD卡在手机上,这样MountService就可以成功创建ASEC文件在sd卡上,所以应用安装成功。遇到此类问题请联系工作人员。

# 参考文献

• http://blog.csdn.net/lilian0118/article/details/24455019

# 12.2.4. INSTALL_FAILED_DEXOPT

# 问题分析

ADT和SDK Tool在将source转化成apk的时候因为dex优化失败,导致方法数超标进而抛出异常(类中的每一个方法都分配有一个id,字节码中以ID标识和调用方法)。

#### 解决方案

因为方法数超标,建议采用下述方式:

• 检查代码,删出无用jar包和代码,尤其是自动生成的get/set和无用的类。

• 将部分Java代码封装到JNI中。

# 12.2.5. INSTALL_FAILED_MEDIA_UNAVAILABLE

# 问题分析

Android应用安装位置不可用。应用安装位置有两个:ROM、sdcard。如果严格指定安装到sdcard,但是设备没有sdcard时,就会报此类错误。

# 解决方案

严格指定安装位置到sdcard时,如果设备没有sdcard会报此类错误,建议在 AndroidManifest.xml 中配置 android:installLocation 。其中的auto参数可以指定程 序可以被安装在外部存储介质上(例如: SD Card),但是默认会被安装到手机内存中。当手机内存为空时,程序将被安装到外部存储介质上。当程序安装到手机上后, 用户可以决定把程序放在外部储介质还是内存中。

# 代码示例

<manifestxmlns:android="http://schemas.android.com/apk/res/android"android:installLocation="auto"></mainfest>

# 12.2.6. INSTALL_FAILED_NO_MATCHING_ABIS

# 问题分析

当安装的App包含native libraries(一般是so文件)的时候,如果没有对应机器CPU架构的库文件,就会出现这种问题。比如,开发者编译了一个armv7平台的App,但是 想要装在intel架构的设备上,就会出现这个错误。

### 解决方案

安装的App包含Native Libraries(一般是so文件)的时候,因为没有对应机器CPU架构的库文件,就会报出此类问题。建议在编译App时,先用NDK编译出相应的CPU架构 的库文件。

步骤一:编写 Application.mk 文件在jni目录下(例如,即hello-jni.c同级目录下)新建一个 Application.mk 文件,在文件中添加如下代码:

APP_ABI := armeabi armeabi-v7a x86

这段话表示你可以同时生成三个处理器的so库。如果没有或不想使用 Application.mk 文件,则在ndk-build参数中添加如下代码:

APP_ABI="armeabi armeabi-v7a x86 mips"

#### 即运行:

ndk-build APP_ABI="armeabi armeabi-v7a x86 mips"

步骤二:生成.so共享库文件 Android.mk 文件已经编写好了,现在可以用Android NDK开发包中的ndk-build脚本生成对应的.so共享库了,方法如下:

mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ ls
AndroidManifest.xml assets bin default.properties gen jni libs obj res src
mqc@ubuntu:~/workspace/android/NDK/hello-jni\$ ndk-build
Gdbserver : [arm-linux-androideabi-4.4.3] libs/armeabi/gdbserverGdbsetup : libs/armeabi/gdb.setupInstall : libhello-jni.so => l
ibs/armeabi/libhello-jni.so

这样,在文件夹 hello-jni/libs 下生成了三个文件夹armeabi/armeabi-v7a/x86,里面已经正确的生成了 libhello-jni.so 共享库了。

步骤三:在eclipse重新编译HelloJni工程,生成apkeclipse中刷新下HelloJni工程,重新编译生成apk,libhello-jni.so共享库会一起打包在apk文件内。

# 12.2.7. INSTALL_FAILED_UID_CHANGED

### 问题分析

应用在上次卸载时,由于应用中的native程序未终止运行而占用 /data/data/{packageName} 目录,导致 /data/data/ 目录下的App相关内容未被删除,多数情况下

能在logcat中找到报错: Installer: rm -rf failed, directory is not empty 。再次安装时无法覆盖,导致报出此类问题。

### 解决方案

此时如果设备已经Root,则直接删除 /data/data/ 目录下的应用包名文件夹;如果没有root,需要将手机恢复出厂设置。此外,开发者要额外注意容易出现此类问题的 手机,对此类设备的用户给出提示。用户无需操心,阿里移动测试平台会帮助解决。

# 12.2.8. INSTALL_FAILED_VERIFICATION_TIMEOUT

#### 问题分析

部分机型在使用adb命令安装应用时,设备会弹窗询问用户是否允许安装,假如一段时间内(一般是30s),如果没有点击确认弹窗此类设备会阻止安装,并提示用户: INSTALL_FAILED_VERIFICATION_TIMEOUT。

# 解决方案

此时需要点击确认安装的按钮。如果报此类异常,请联系技术支持。

# 12.2.9. INSTALL_FAILED_VERSION_DOWNGRADE

# 问题分析

设备中已经安装了此App的更高版本,无法降级安装。

# 解决方案

建议先卸载之前的版本,再安装此版本。

# 12.2.10. INSTALL_PARSE_FAILED_INCONSISTENT_CERTIFICATES

# 问题分析

由于App签名冲突造成。如果设备上已经安装了其他签名的相同包名App,再安装其他签名的就会报出此类错误。

# 解决方案

- 更换签名文件,重新签名。
- 更改应用的包名,避免冲突。
### 12.2.11. INSTALL_PARSE_FAILED_NO_CERTIFICATES

### 问题分析

应用中没有签名信息时安装失败。

#### 解决方案

因为没有签名信息导致安装失败,建议使用Eclipse或者Android Studio对项目重新编译打包。因为应用有可能类名,包名相同,所以签名作为区分作用,保证安装时签名 不同的包不被替换。APK如果使用一个key签名,发布时另一个key签名的文件将无法安装或覆盖老的版本,这样可以防止你已安装的应用被恶意的第三方覆盖或替换掉。 签名其实也是开发者的身份标识,是开发者自己产生的数字证书,即所谓的自签名。具体的签名过程请参考文献。

### 参考文献

• https://developer.android.com/studio/publish/app-signing.html (官方签名文档)

http://blog.csdn.net/fengye810130/article/details/9159419(签名文档)

## 12.3. 启动时错误

# 12.4. Native_crash

### 问题分析

Native的Crash是指在C/C++运行时出错,系统产生了Linux错误信号,导致的进程出错退出。可以通过系统的logcat来分析Crash日志。

### 解决方案

应用在C/C++运行时出错,系统产生了Linux错误信号,导致的进程出错退出。建议开发者使用NDK工具ndk-stack进行分析定位。 Android开发中,在Java层可以方便的捕获crashlog,但对于Native层的crashlog通常无法直接获取,只能通过系统的logcat来分析Crash日志。这里我们建议开发者使用 NDK工具ndk-stack进行分析定位。

- 1. 下载最新版NDK。ndk-stack工具就在NDK主目录下。
- 2. 在mqc管理中心找到出现Native Crash的测试,并下载日志。
- 3. 根据失败机型查找该机型的CPU信息 ro.product.cpu.abi ,如图三星N7100的CPU类型为armeabi-v7a。(目前mqc上的模拟器CPU类型均为x86_64。)
- 4. 使用ndk-stack分析出错位置,命令格式如下:

\$NDK/ndk-stack -sym \$PROJECT_PATH/obj/local/\$cpu.abi -dump \$LOGCAT_PATH

#### 本示例使用的命令如下:

ndk-stack -sym ./workspace2/testNdkStack/obj/local/armeabi-v7a/ -dump ./logcat.log > result.log

在result.log中可以分析定位到出现该Crash的对应代码文件和具体行数。