百度智能云

ARMCM 文档



【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。 未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。 本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

【免责声明】

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。 如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

目录

目录 2
产品动态 3
镜像发布历史 3
新功能发布历史 21
产品介绍 30
什么是云手机实例 31
产品优势 31
应用场景 31
云直播 31
安全办公云手机 32
实例规格 32
通用部分 32
什么是云手机（速享版） 33
云手机（速享版）接入指南 33
快速入门 34
快速入门流程 34
购买云手机实例 35
购买营销套件服务 38
产品定价 39
技术文档 41
云手机Android SDK 41
OPEN API参考 60
云手机Web SDK 318
相关协议&政策 326
服务等级协议SLA 326
百度云手机软件开发工具包隐私政策 327
常见问题 336
使用类 336
客户常见问题 336
为什么不能安装、卸载、更新APP？ 336
是否可以安装应用？在哪里安装应用？ 336
为什么APP会安装失败？ 336
云手机支持远程adb调用吗？ 336
云手机可否支持绑定SIM卡？云手机能否收到登录短信？ 336
一台电脑同时登录多台云手机，这些云手机的IP是否一样？ 337
是否可以批量安装APP？ 337
需要海外机房怎么办？ 337

产品动态

镜像发布历史

新功能发布：2026-10-30

镜像ID	软件包类型	镜像名称	功能概述
1291479154590888003	整包	3588_aosp12_v3.64.0(2026-09-22)	1. 修复读取GPU load信息时概率性触发系统崩溃的问题
7668576550810982035	自定义		
1291479154590887999	整包	3588_aosp15_v1.31.0(2026-09-22)	1. 修复重复多次打开关闭摄像头导致HAL崩溃问题
3921581654257808058	自定义		

新功能发布：2026-09-24

镜像ID	软件包类型	镜像名称	功能概述
3498225252416716936	整包	3588_aosp10_v2.59.0(2026-08-25)	1. 内置应用支持按包名卸载 2. 支持接收游戏手柄上的顶头按键信息 3. DuFS相关功能优化
3417168522753545847	自定义	示范： 服务器类型：3588 系统版本：安卓10 版本号：2.59.0	
8587292877067907208	整包	3588_aosp12_v3.62.0(2026-08-25)	1. 内置应用支持按包名卸载
3417168522753556716	自定义		
3498225252416716940	整包	3588_aosp15_v1.30.0(2026-08-25)	1. 内置应用支持按包名卸载 2. 短信弹窗通知优化（仅整包功能，自定义包没有）
3417168522753553867	自定义		
1291479154590887983	整包	8550_aosp13_v1.50.0(2026-09-17)	1. 修复镜像挂载问题
7668576550810597408	自定义		
3498225252416716934	整包	845_aosp10.0_v1.56.0(2026-08-25)	1. 内置应用支持按包名卸载
3146952545111194533	自定义		
3498225252416716937	整包	845_aosp8.1_v2.46.0(2026-08-25)	1. 内置应用支持按包名卸载
3146952545111196325	自定义		

新功能发布：2026-08-27

镜像ID	软件包类型	镜像名称	功能概述
3498225252416716826	整包	3588_aosp10_v2.57.0(2026-07-28) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.57.0	1. 内存管理优化 2. 支持模拟真实点击
1237418266293322043	自定义		
8587292877067907132	整包	3588_aosp12_v3.61.0(2026-08-11)	1. Android12 代码库升级（仅整包功能，自定义包没有） 2. 内存管理优化 3. 支持模拟真实点击 4. 解决摄像头变形问题 5. 制作应用镜像进度通知优化 6. 对象存储增加重连机制 7. 修复对象存储还原空间大小问题 8. 支持接收游戏手柄上的顶头按键信息
1237418266294335858	自定义		
8587292877067907134	整包	3588_aosp15_v1.29.0(2026-08-11)	1. 修复某些场景下视频通话无画面的问题 2. 安全漏洞修复 3. 支持接收游戏手柄上的顶头按键信息
1237418266294336897	自定义		
3498225252416716827	整包	8550_aosp13_v1.49.0(2026-07-28)	1. 支持模拟真实点击
1237418266293323285	自定义		
3498225252416716863	整包	845_aosp10.0_v1.55.0(2026-08-11)	1. 安全漏洞修复 2. 支持接收游戏手柄上的顶头按键信息 3. 支持模拟真实点击
8371120076049984566	自定义		
3498225252416716865	整包	845_aosp8.1_v2.45.0(2026-08-11)	1. 安全漏洞修复 2. 支持接收游戏手柄上的顶头按键信息 3. 支持模拟真实点击
1237418266294343487	自定义		

新功能发布：2026-07-30

镜像ID	软件包类型	镜像名称	功能概述
6092281991015145555	整包	3588_aosp10_v2.55.0(2026-06-30) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.55.0	1. 软件包支持F2FS文件系统
8217982735898706561	自定义		
6092281991015145624	整包	3588_aosp12_v3.59.0(2026-07-14)	1. 支持摄像头参数仿真（仅整包功能，自定义包没有） 2. 代码库升级，对齐芯片最新版本SDK（仅整包功能，自定义包没有） 3. 软件包支持F2FS文件系统
8217982735899776144	自定义		
6092281991015145556	整包	3588_aosp15_v1.27.0(2026-06-30)	1. 软件包支持F2FS文件系统
1723792073230682749	自定义		
2453373447100645466	整包	8550_aosp13_v1.47.0(2026-06-30)	1. 软件包支持F2FS文件系统 2. 支持标识符仿真，虚拟机每次重启后均会更新
8217982735898709286	自定义		

新功能发布：2026-06-29

镜像ID	软件包类型	镜像名称	功能概述
6092281991015145477	整包	3588_aosp10_v2.54.0(2026-06-03) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.54.0	1. 支持点击屏幕参数自定义 2. agent支持返回swap配额、内存配额、存储配额 3. 修复 Android10 迷你枪战精英media.codec模块crash问题（仅整包功能，自定义包没有）
4047633279315999489	自定义		
6092281991015145476	整包	3588_aosp12_v3.56.0(2026-06-03)	1. 支持点击屏幕参数自定义 2. agent支持返回swap配额、内存配额、存储配额 3. 修复 Android10 迷你枪战精英media.codec模块crash问题（仅整包功能，自定义包没有）
4047633279315998843	自定义		
2453373447100645379	整包	3588_aosp15_v1.25.0(2026-06-03)	1. 修复图库编辑闪退（仅整包功能，自定义包没有） 2. 提高快照还原成功率 3. 支持点击屏幕参数自定义 4. agent支持返回swap配额、内存配额、存储配额
4047633279315991676	自定义		
2453373447100645453	整包	845_aosp10.0_v1.53.0(2026-06-25)	1. 提高快照还原成功率 2. 支持点击屏幕参数自定义 3. agent支持返回swap配额、内存配额、存储配额
8217982735898342970	自定义		
2453373447100645385	整包	845_aosp8.1_v2.43.0(2026-06-03)	1. 提高快照还原成功率 2. 支持点击屏幕参数自定义 3. agent支持返回swap配额、内存配额、存储配额 4. 设备集成openssl命令（仅整包功能，自定义包没有）
1903919855095600888	自定义		

新功能发布：2026-05-29

镜像ID	软件包类型	镜像名称	功能概述
2624495781440024598	整包	3588_aosp10_v2.52.0(2026-04-29)	1. 解决二重螺旋绿条纹问题（仅整包功能，自定义包没有） 2. 分辨率修改接口功能优化 3. 支持apks格式的应用预置安装和预置卸载 4. DuFS支持block模式（仅整包功能，自定义包没有） 5. iops限速支持自适应配置
1903919855093215188	自定义	示范： 服务器类型：3588 系统版本：安卓10 版本号：2.52.0	
2624495781440024599	整包	3588_aosp12_v3.54.0(2026-04-29)	1. Android12 修复剪映卡顿丢帧的问题 2. 分辨率修改接口功能优化 3. 支持apks格式的应用预置安装和预置卸载 4. DuFS支持block模式（仅整包功能，自定义包没有） 5. iops限速支持自适应配置
1903919855093222972	自定义		
2624495781440024606	整包	3588_aosp15_v1.24.1(2026-04-30)	1. 修复SELinux内存泄漏的问题 2. 分辨率修改接口功能优化 3. 支持apks格式的应用预置安装和预置卸载 4. DuFS支持block模式（仅整包功能，自定义包没有） 5. 支持企微双向录音（仅整包功能，自定义包没有） 6. iops限速支持自适应配置
4047633279313683915	自定义		
7407302671533773147	整包	8550_aosp13_v1.46.0(2026-04-16)	1. 分辨率修改接口功能优化 2. 支持apks格式的应用预置安装和预置卸载
6641691358571357693	自定义		
7407302671533773145	整包	845_aosp10.0_v1.52.0(2026-04-15)	1. iops限速支持自适应配置
6641691358571301031	自定义		
7407302671533773144	整包	845_aosp8.1_v2.42.0(2026-04-15)	1. 支持apks格式的应用预置安装和预置卸载 2. iops限速支持自适应配置
8037807243056444929	自定义		

新功能发布：2026-04-28

镜像ID	软件包类型	镜像名称	功能概述
7407302671533773115	整包	3588_aosp10_v2.50.0(2026-04-01) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.50.0	1. 优化connlimit脚本和解决gnss hal崩溃问题（仅整包功能，自定义包没有） 2. 通用软件包Swap空间大小支持调整 3. 解决新设备软件包刷机或设备重置后壁纸和机型出错的问题 4. 解决修改dpi分辨率后点击屏幕坐标出错的问题（仅整包功能，自定义包没有） 5. 提高逻辑分辨率设置上限（仅整包功能，自定义包没有）
8037807243055618566	自定义		
7407302671533773117	整包	3588_aosp12_v3.52.1(2026-04-01)	1. 优化connlimit脚本和解决gnss hal崩溃问题（仅整包功能，自定义包没有） 2. 通用软件包Swap空间大小支持调整 3. 解决新设备软件包刷机或设备重置后壁纸和机型出错的问题 4. 解决修改dpi分辨率后点击屏幕坐标出错的问题（仅整包功能，自定义包没有） 5. 提高逻辑分辨率设置上限（仅整包功能，自定义包没有） 6. 解决使用安卓10的内核时，ls -al /stroage/emulated提示没权限的问题（仅整包功能，自定义包没有）
8037807243055625875	自定义		
6479561150704263447	整包	3588_aosp15_v1.22.0(2026-03-25)	1. 修改webview配置，避免沙盒进程访问root（仅整包功能，自定义包没有） 2. 补充完善xminitor上报信息（仅整包功能，自定义包没有） 3. 通用软件包Swap空间大小支持调整 4. 解决新设备软件包刷机或设备重置后壁纸和机型出错的问题
8037807243055244645	自定义		
6479561150704263458	整包	8550_aosp13_v1.45.0(2026-03-26)	1. 通用软件包Swap空间大小支持调整
8037807243055315900	自定义		
7407302671533773082	整包	845_aosp10.0_v1.51.0(2026-03-26)	1. 解决修改dpi分辨率后点击屏幕坐标出错的问题（仅整包功能，自定义包没有） 2. 提高逻辑分辨率设置上限（仅整包功能，自定义包没有） 3. 通用软件包Swap空间大小支持调整
8037807243055312946	自定义		
6479561150704263457	整包	845_aosp8.1_v2.41.0	1. 提高逻辑分辨率设置上限（仅整包功能，自定义包没有）

8037807243055313473	自定义	(2026-03-26)	2. 通用软件包Swap空间大小支持调整

新功能发布：2026-03-04

镜像ID	软件包类型	镜像名称	功能概述
9046586396044484654	整包	3588_aosp10_v2.48.0(2026-01-19) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.48.0	1. dufs快照制作增加baseSnapshot用于记录快照关联关系 2. dufs快照卸载CDN增加环境校验 3. 修正dufs在线快照超时时间参数
7092024083090801262	自定义		
248934533859328828 3	整包	3588_aosp12_v3.50.1(2026-01-28)	1. dufs快照制作增加baseSnapshot用于记录快照关联关系 2. dufs快照卸载CDN增加环境校验 3. 修正dufs在线快照超时时间参数 4. dufs在游戏资源加载速度慢的问题优化
3155878008769722731	自定义		
248934533859328825 1	整包	3588_aosp15_v1.20.0(2026-01-19)	1. dufs快照制作增加baseSnapshot用于记录快照关联关系 2. dufs快照卸载CDN增加环境校验 3. 修正dufs在线快照超时时间参数
7092024083090800442	自定义		
9046586396044484652	整包	8550_aosp13_v1.43.0(2026-01-19)	1. dufs快照制作增加baseSnapshot用于记录快照关联关系 2. dufs快照卸载CDN增加环境校验 3. 修正dufs在线快照超时时间参数
3155878008769144359	自定义		
9046586396044484691	整包	845_aosp8.1_v2.40.0(2026-02-03)	1. 解决容器启动后发送命令超时问题
3155878008770100184	自定义		

新功能发布：2026-01-30

镜像ID	软件包类型	镜像名称	功能概述
2489345338593288227	整包	3588_aosp10_v2.47.1(2026-01-04)	1. 开启CONFIG_ANON_VMA_NAME，解决部分APP检查maps导致的风控问题 2. 关闭PCIe总线省电模式，减少 essd/ssd hung task 概率
3155878008768420318	自定义	示范： 服务器类型：3588 系统版本：安卓10 版本号：2.47.1	
9046586396044484634	整包	3588_aosp12_v3.49.1(2026-01-04)	1. 开启CONFIG_ANON_VMA_NAME，解决部分APP检查maps导致的风控问题 2. 关闭PCIe总线省电模式，减少 essd/ssd hung task 概率 3. 与安卓10版本统一内核
7092024083090073561	自定义		
9091612527662333990	整包	3588_aosp15_v1.18.0(2025-12-17)	1. 关闭云手机休眠功能，避免长时间不操作导致熄屏的问题（仅整包功能，自定义包没有） 2. 镜像修复HWC崩溃问题（仅整包功能，自定义包没有）
5347724527165010 51	自定义		
1948902705626624048	整包	8550_aosp13_v1.41.0(2025-12-18)	1. 内置xmonitor（仅整包功能，自定义包没有） 2. 关闭云手机休眠功能，避免长时间不操作导致熄屏的问题（仅整包功能，自定义包没有） 3. 内核修复namespace泄漏问题
5347724527165342 31	自定义		
2489345338593288245	整包	845_aosp10_v1.50.0(2025-01-16)	1. 镜像修复HWC崩溃问题（仅整包功能，自定义包没有）
7092024083090680022	自定义		

新功能发布：2025-12-30

镜像ID	软件包类型	镜像名称	功能概述
5758933512559136873	整包	3588_aosp10_v2.44.0(2025-11-18) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.44.0	1. 提供在线快照接口（仅整包功能，自定义包没有） 2. 修复容器设置界面内存显示不准问题（仅整包功能，自定义包没有） 3. 修复Android10 RF韩服登录后在剧情动画黑屏等问题 4. 修复AV1 格式解码错误导致物理机重启的问题
3363018584196960292	自定义		
5407652812278063197	整包	3588_aosp12_v3.46.0(2025-11-18)	1. 支持可配摄像头分辨率（仅整包功能，自定义包没有） 2. 优化iaas agent lxc查询IP逻辑 3. 优化新版软件包容器设置界面内存显示不准问题 4. 修复remote-play回调bug 5. 修复AV1 格式解码错误导致物理机重启的问题（仅整包功能，自定义包没有）
4038558531095555302	自定义		
5407652812278063198	整包	3588_aosp15_v1.16.0(2025-11-18)	1. Android15代码升级，对齐官方SDK版本（仅整包功能，自定义包没有） 2. 支持可配摄像头分辨率（仅整包功能，自定义包没有） 3. 优化iaas agent lxc查询IP逻辑 4. 修复容器设置界面内存显示不准问题 5. 修复Android15 批量重置后虚拟机概率性无法启动等问题（仅整包功能，自定义包没有）
4038558531095555370	自定义		
5407652812278063196	整包	8550_aosp13_v1.39.0(2025-11-18)	1. 优化iaas agent lxc查询IP逻辑 2. 修复容器设置界面内存显示不准问题 3. 修复内置相机camera2拍照生成图片绿边问题（仅整包功能，自定义包没有）
3363018584196960046	自定义		
5758946306585890818	整包	845_aosp10_v1.48.0(2025-11-18)	1. 提供在线快照接口 2. 修复容器设置界面内存显示不准问题
3363018584196987577	自定义		
5758946306585890817	整包	845_aosp8.1_v2.37.0(2025-11-18)	1. 提供在线快照接口 2. 修复容器设置界面内存显示不准问题
4038558531095564445	自定义		

新功能发布：2025-11-27

镜像ID	软件包类型	镜像名称	功能概述
57589335125591 36805	整包	3588_aosp10_v2.42.0(2025-10-22) 示范： 服务器类型：3588 系统版本：安卓10 版本号：2.42.0	1. 新增在线快照接口 2. 修复软件包容器设置界面内存显示不准的问题 3. 修复RF韩服登录后剧情动画黑屏等问题（仅整包功能，自定义包没有） 4. 修复AV1格式解码错误导致物理机重启的问题（仅整包功能，自定义包没有）
33630185841962 91197	自定义		
57589335125591 36806	整包	3588_aosp12_v3.44.0(2025-10-22)	1. 修复dufs概率性内存crash的问题 2. 优化lxc IP返回逻辑 3. 修复软件包容器设置界面内存显示不准的问题 4. 修复remote-play回调报错 5. 支持可配摄像头分辨率（仅整包功能，自定义包没有） 6. 修复AV1格式解码错误导致物理机重启的问题（仅整包功能，自定义包没有）
33630185841962 91258	自定义		
57589335125591 36808	整包	3588_aosp15_v1.14.0(2025-10-22)	1. 优化lxc IP返回逻辑 2. 修复软件包容器设置界面内存显示不准的问题 3. 修复批量重置后虚拟机概率性无法启动等问题 4. 支持可配摄像头分辨率（仅整包功能，自定义包没有） 5. 代码升级，对齐官方SDK 版本（仅整包功能，自定义包没有）
33630185841962 92082	自定义		
57589335125591 36809	整包	8550_aosp13_v1.37.0(2025-10-22)	1. 优化lxc IP返回逻辑 2. 修复软件包容器设置界面内存显示不准的问题 3. 修复remote-play回调报错 4. 优化内置相机camera2拍照生成图片绿边的情况（仅整包功能，自定义包没有）
33630185841962 92689	自定义		
57589335125591 36811	整包	845_aosp10_v1.46.0(2025-10-23)	1. 新增在线快照接口 2. 修复软件包容器设置界面内存显示不准的问题
40385585310948 89791	自定义		

5758933512559136807	整包	845_aosp8.1_v2.35.0(2025-10-23)	1. 新增在线快照接口 2. 修复软件包容器设置界面内存显示不准的问题
3363018584196291722	自定义		

新功能发布：2025-10-30

镜像ID	镜像名称	功能概述
3642224735359033378	8550_aosp13_v1.36.0(2025-09-18)	1. 设备支持动态渲染帧率设置 2. 一键新机功能文件路径优化 3. 盒子软件包支持单实例更新 4. 修复iaas agent版本升级导致实例imei被修改的问题
3642224735359033406	3588_aosp15_v1.12.0(2025-09-18)	1. 3588设备SMT编码“高码率模糊”问题修复 2. 盒子更换SSD无需刷机 3. 设备支持动态渲染帧率设置 4. 一键新机功能文件路径优化 5. 盒子软件包支持单实例更新 6. 修复iaas agent版本升级导致实例imei被修改的问题 7. 安卓15镜像增加应用分屏能力
3642224735359033401	3588_aosp12_v3.42.0(2025-09-18)	1. 3588设备SMT编码“高码率模糊”问题修复 2. 盒子更换SSD无需刷机 3. dg版本升级 4. 设备支持动态渲染帧率设置 5. 一键新机功能文件路径优化 6. 盒子软件包支持单实例更新 7. 修复iaas agent版本升级导致实例imei被修改的问题
		1. 3588设备SMT编码“高码率模糊”问题修复 2. 盒子更换SSD无需刷机 3. dg版本升级

3642224735359033402	3588_aosp10_v2.40.0(2025-09-18)	4. 设备支持动态渲染帧率设置 5. 一键新机功能文件路径优化 6. 盒子软件包支持单实例更新 7. 修复iaas agent版本升级导致实例imei被修改的问题
9046544265866543148	845_aosp10_v1.45.0(2025-09-18)	1. 修复接口截图失败的问题 2. manage-agent状态异常问题优化 3. 盒子软件包支持单实例更新 4. 修复iaas agent版本升级导致实例imei被修改的问题
9046544265866543147	845_aosp8.1_v2.34.0(2025-09-18)	1. 修复接口截图失败的问题 2. 盒子软件包支持单实例更新 3. 修复iaas agent版本升级导致实例imei被修改的问题 4. 盒子更换SSD无需刷机 5. Android 8 默认开启安装残留文件清理

新功能发布：2025-09-05

镜像ID	镜像名称	功能概述
7974677675551047700	8550_aosp13_v1.33.0(2025-08-05)	1. 摄像头推流切无人直播问题修复 2. remote play-HWC推流优化 3. 存在socket阻止文件时禁止访问虚机的data目录 4. 支持容器粒度流量上报 5. manage-agent状态异常问题优化 6. 优化sed命令导致异常的情况
		1. remote play-HWC推流优化 2. 存在socket阻止文件时禁止访问虚机的data目录 3. 支持容器粒度流量上报

7974677675551047699	3588_aosp15_v1.9.0(2025-08-05)	4. 升级dg-v5.14 5. 优化libcore兼容性 6. 设备启动后应用安装逻辑优化 7. 解决webrtc-p2p模式avtrans接口无法关闭声音问题 8. manage-agent状态异常问题优化 9. 优化sed 命令导致异常的情况
7974677675551047705	3588_aosp12_v3.39.0(2025-08-08)	1. remote play-HWC推流优化 2. 存在socket阻止文件时禁止访问虚机的data目录 3. 支持容器粒度流量上报 4. 升级dg-v5.14 5. 优化libcore兼容性 6. 设备启动后应用安装逻辑优化 7. 解决webrtc-p2p模式avtrans接口无法关闭声音问题 8. 内核内存隔离显示优化 9. manage-agent状态异常问题优化 10. 优化sed命令导致异常的情况
5443654684968914978	3588_aosp10_v2.37.0(2025-08-05)	1. remote play-HWC推流优化 2. 存在socket阻止文件时禁止访问虚机的data目录 3. 支持容器粒度流量上报 4. Android10编解码库升级 5. 内核内存隔离显示优化 6. manage-agent状态异常问题优化 7. 优化sed命令导致异常的情况 8. 云机支持企微双向录音

新功能发布：2025-07-25

镜像ID	镜像名称	功能概述
4903212516313165826	8550_aosp13_v1.31.0(2025-07-01)	1. DUFS支持硬链接 2. 虚机系统空间占用显示支持可配置 3. 快照挂载逻辑优化 4. p2p模式推流相关优化 5. 开放接口相关优化
1003094866958036995	3588_aosp15_v1.7.0(2025-07-01)	1. 适配DUFS以及DUFS支持硬链接 2. 虚机系统空间占用显示支持可配置 3. p2p模式推流相关优化 4. manage-agent崩溃等问题修复 5. 开放接口相关优化
1003094866958036994	3588_aosp12_v3.37.0(2025-07-01)	1. dufs-cli日志支持写入日志文件 2. 挂载脚本支持recover指定uid/gid 3. 安卓12集成拨号APP 4. DuFS支持硬链接 5. 虚机系统空间占用显示支持可配置 6. p2p模式推流相关优化
1003094866958036996	3588_aosp10_v2.35.0(2025-07-01)	1. dufs-cli日志支持写入日志文件 2. 虚机系统空间占用显示支持可配置 3. DuFS支持硬链接 4. p2p模式推流相关优化 5. manage-agent崩溃等问题修复

新功能发布：2025-05-29

镜像ID	镜像名称	功能概述
4290700291271323657	8550_aosp13_v1.28.0(2025-04-29)	1. 支持摄像头麦克风回调和抢占功能 2. 支持云机对拷 3. 优化云游戏广播通知 4. 优化remote跟libbac的使用逻辑 5. 修复webrtc-p2p模式下推流偶现卡死问题
4083535031286300703	3588_aosp15_v1.3.0(2025-05-20)	1. 升级dg版本v5.13 2. 更新dgav模块支持摄像头麦克风 3. 更新安卓15机型库 4. 支持platform改机参数 5. 支持虚拟机特征屏蔽 6. 优化remote跟libbac的使用逻辑 7. 修复H5本地输入法输入问题
4083535031286300679	3588_aosp12_v3.32.0(2025-04-29)	1. 新增仿真服务授权管理功能3 2. 支持摄像头麦克风回调和抢占功能 3. 支持云机对拷 4. 优化截图流适配VPC 5. 优化remote跟libbac的使用逻辑 6. 优化设备网络会话数量限制 7. 修复webrtc-p2p模式下推流偶现卡死问题
4290700291271323658	3588_aosp10_v2.29.1(2025-04-30)	
4083535031286300675	845_aosp10_v1.37.0(2025-04-29)	
4290700291271323653	845_aosp8.1_v2.27.0(2025-04-29)	

发布时间：2025-04-14

镜像ID	镜像名称	功能概述
507666707417677828	3588_aosp15_v1.0.0(2025-04-02)	1. 新系统发布，支持最新安卓15
8452004025491873823	8550_aosp13_v1.24.1(2025-03-13)	1. 升级dg5.13 2. 优化DuFS清理逻辑以及文件校验
4488836397410750482	3588_aosp12_v3.28.1(2025-03-13)	
4488836397410750477	3588_aosp10_v2.25.1(2025-03-13)	
4488836397410750483	845_aosp10_v1.33.1(2025-03-13)	
507666707417677838	845_aosp8.1_v2.25.0(2025-04-02)	1. 支持后台仿真服务授权管理功能 2. 优化配置导致虚拟机故障问题

发布时间：2025-03-14

镜像ID	镜像名称	功能概述
4488836397410750467	88550_aosp13_v1.22.0(2025-02-24)	1. 优化虚拟机启动逻辑及事件 2. 优化虚拟机通讯链路 3. 优化快照挂载状态 4. 优化快照卸载超时处理 5. 优化恢复出厂设置接口，解决偶现失败的情况 6. 支持摄像头1080P分辨率 7. 支持基于对象存储的备份还原
8452004025491873796	3588_aosp12_v3.26.0(2025-02-24)	
4488836397410750465	3588_aosp10_v2.23.0(2025-02-24)	
4488836397410750468	845_aosp10_v1.31.0(2025-02-24)	
4488836397410750469	845_aosp8.1_v2.23.0(2025-02-24)	

发布时间：2025-02-14

镜像ID	镜像名称	功能概述
6479414955299864588	8550_aosp13_v1.19.0(2025-01-06)	1. 优化设备DuFS开关 2. lmkd支持psi cgroupV1 3. 升级镜像GBoard输入法为64位
4930163260468854811	3588_aosp12_v3.22.0(2025-01-06)	1. 优化设备DuFS开关 2. 修复三键导航sdk控制开关 3. 升级dg5.12
4930163260468854810	3588_aosp10_v2.20.0(2025-01-06)	1. 优化设备DuFS开关 2. 修复三键导航sdk控制开关 3. 升级dg5.12

发布时间：2025-01-17

镜像ID	镜像名称	功能概述
4930163260468854788	3588_aosp12_v3.20.0(2024-12-16)	1. 优化IaaS软件模块信息采集规范 2. 修复同物理机并发内置应用问题 3. 优化sim配置、一键新机融合、proxy更新
561671636425011202	3588_aosp10_v2.18.0(2024-12-16)	1. 优化IaaS软件模块信息采集规范 2. 优化一键新机融合优化数据同步逻辑 3. 优化ROOT unix domain socket访问权限 4. 优化installApp创建目录的权限 5. 支持系统镜像SDK开启/关闭三键导航 6. 支持云手机设置指定dns 7. 修复mgr内置接口问题
561671636425011216	845_aosp10_v1.28.0(2024-12-27)	1. 修复mgr内置接口问题 2. 支持系统镜像SDK开启/关闭三键导航 3. 支持云手机设置指定dns 4. 修复重置后截图不实时
4930163260468854797	845_aosp8.1_v2.20.0(2024-12-25)	1. 支持禁用外网 2. 支持软件包按模块更新 3. 优化IaaS软件模块信息采集规范 4. 优化一键新机融合优化数据同步逻辑 5. 支持镜像定时服务 6. 优化ROOT unix domain socket访问权限 7. 支持云手机设置指定dns 8. 支持系统镜像SDK开启/关闭三键导航 9. 修复mgr内置接口问题

发布时间：2024-12-03

镜像ID	镜像名称	功能概述
63532728314396754l5	3588_aosp12_v3.14.0(2024-10-29)	1. 去除个别模块，防止识别为异常环境。 2. 内置proxy插件更新，优化dns设置和特殊语言启动失败问题。 3. 修复dg dump location中gps显示异常问题。 4. 支持network/fuse location定位模拟。 5. 解决aosp10状态栏在重置后首次启动无法完整下拉的问题。
6353272831439675406	3588_aosp10_v2.11.0(2024-10-14)	

发布时间：2024-10-29

镜像ID	镜像名称	功能概述
3128683890169999368	3588_aosp10_v2.9.0(2024-09-10)	1. 限速、存储隔离问题优化 2. 日志管理优化；规范软件日志存放路径 3. DuFS状态优化
6848655575367237645	3588_aosp12_v3.10.0(2024-09-10)	1. Dufs状态优化；支持取消备份；错误码优化 2. 推流问题优化；日志规范化；ROOT开关优化
4011358924465319941	845_aosp8.1_v2.9.0(2024-06-18)	1. 限速功能、存储隔离优化 2. 优化版本记录

新功能发布历史

2026年9月

平台	变更模块	变更详情	跳转链接
aPaaS	实例运营	实例运营页增加「检测释放」功能，支持客户自助恢复释放失败实例，提升设备回收效率。	实例运营
	套餐管理	商品套餐管理页支持 Excel 导出，便于查阅全部套餐并整理对外材料。	套餐管理
	仿真授权管理	营销套件授权码新增支持换绑实例功能及接口，辅助客户提升资源利用率。	仿真授权管理
	API功能	Operator支持OpenAPI与SDK接入，提升接入灵活性与扩展性。	Operator操作
		新增「应用名单管理」系列接口，满足客户对下游用户安装应用的精细化管控需求。	应用名单管理
云手机速享版/SDK	云手机速享版	H5客户端UI整体升级，提升用户体验。	/
		PC客户端新增「批量粘贴文本」功能，提升文本录入效率。	/

2026年8月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	批量控制新增支持从控机单独操作，客户无需退出可即时处理从控设备，提升运营效率。	实例列表
	实例运营	实例运营页增加「检测释放」功能，支持客户自助恢复释放失败实例，提升设备回收效率。	实例运营
	API功能	新增「应用名单管理」系列接口，满足客户对下游用户安装应用的精细化管控需求。	分页查询应用名单
		「实例汇总数据查询」接口升级，支持多状态实例数量统计。	客户实例汇总数据查询
云手机速享版/SDK	云手机速享版	PC、安卓客户端UI整体升级，提升用户体验。	/
		PC、安卓客户端打包支持主题色选择，提升客户灵活性。	/

2026年7月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	实例列表支持单页查询万级实例信息及批量操作能力，提升运维效率。	实例列表
		实例列表-日志详情支持客户按单实例监控性能，帮助业务精细化运营。	实例列表
	工作流	工作流列表新增任务发起人的账号信息，便于客户自行追查操作历史。	工作流
	快照创建记录	aPaaS支持查询和导出快照创建记录，支持客户运营需求。	快照创建记录
	风险应用告警记录	aPaaS新增风险应用自动检测能力，提升客户合规管理效率。	风险应用告警记录
	API功能	「新增实例到期时间」接口优化，支持对已授权的实例进行续时。	新增实例到期时间
		aPaaS提供按包名卸载预置应用接口，方便客户批量卸载同一应用。	预置应用卸载
百度智能云官网	云手机购买	云手机购买页面升级，新增【订单面板】模块，提升自助购买体验与效率。	业务管理
云手机速享版/SDK	云手机速享版	速享版三端客户端增加【使用帮助】入口，提升用户体验。	/
		速享版PC客户端增加【关于我们】页面，便于用户主动检测更新及查看相关协议。	/
	PC SDK	连接云手机新增支持预览流，与控制流形成互补，在多路同屏场景下，更节省带宽成本、更流畅操控体验。	/
	Android SDK	连接云手机新增支持预览流，与控制流形成互补，在多路同屏场景下，更节省带宽成本、更流畅操控体验。	/
	IOS SDK	新增支持云机播放、数据透传、音视频采集和画质设置等接口，支持基于 iOS SDK 搭建原生客户端，满足苹果手机用户登录客户端操控云手机需求。	/

2026年6月

平台	变更模块	变更详情	跳转链接
aPaaS	Operator 配置	新增Operator配置页面，支持界面配置模型和API Key，简化配置流程，提升体验。	Operator 配置
	OpenClaw 升级	云手机OpenClaw支持调用Operator能力，完成设备GUI可视化操控。	/
	API功能	Duapk全新采用物理机本地缓存 + 物理机P2P 分布式分发架构，提升了高并发场景下的应用分发与部署效率。	实例批量安装 duapk 应用镜像
		分页查询物理机信息接口新增出参“物理机运行时长”，便于客户自行排查和定位问题。	分页查询物理机信息
百度智能云官网	云手机产品购买	云手机速享版实现官网在线售卖和自动化交付，提升自助购买体验与效率。	百度云手机（速享版）
云手机速享版/SDK	Android SDK	安卓SDK支持群控功能，为客户提供开箱即用的群控对接接口。	/

2026年5月

平台	变更模块	变更详情	跳转链接
aPaaS	切换账号	新增支持多账号快速切换，不退登快速切换同一个手机号所在的不同商户，提升效率。	/
	应用列表	新增应用安装数的字段信息，方便查询。	应用列表
	实例列表	新增资源包版本、配置项、存储类型、镜像挂载类型、镜像挂载时间的字段信息，方便查询。	实例列表
	OpenClaw配置	新增OpenClaw配置页面，支持界面配置API Key，简化配置流程，提升体验。	OpenClaw配置
	API功能	支持接口查看自动化任务日志，完善脚本自动化功能。	查询自动化任务日志
云手机速享版/SDK	云手机速享版	PC/安卓/H5三端客户端云机列表增加排序功能，提升用户体验。	/
		速享版PC客户端支持打开云手机画面至50个。	/
		PC/安卓/H5三端客户端「云手机续费」体验优化。	/

2026年4月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	新增「应用上架展示名称」字段，支持客户手动修改应用展示名称。	Duapk上架
	应用管理	新增应用白名单功能，支持客户限制下游用户仅能安装白名单内的应用。	应用白名单
		新增「应用镜像」页面，支持客户做页面化的应用镜像查询/管理。	应用镜像管理
	账号管理	角色授权新增“首页数据看板”的功能权限，由客户自行指定账号可见首页各数据板块的功能。	角色管理
	API功能	新增实例预置应用列表接口，支持客户实时查询每个实例已安装的预装应用。	获取实例预置应用安装列表

2026年3月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	文件上传新增支持拖拽上传，提升客户操作体验。	实例列表
		新增实例出口IP及运营商字段，方便非技术人员通过页面自查实例出口IP和运营商信息。	实例列表
	内置应用管理	新增「内置应用管理」页面，支持客户查看内置应用版本记录，支持客户自主打包并更新应用，提升更新效率。	内置应用管理
	带宽明细	支持客户自行查询物理机IP/网段在某个时间点上的带宽明细，提升运营效率。	带宽明细
官网控制台	业务管理	营销套件实现官网在线购买和自动化交付，提高仿真效果，提升业务运行效率。	营销套件购买
		磐玉边缘计算实现官网在线购买，提升自助购买体验与效率。	磐玉边缘计算盒产品介绍

2026年2月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	新增「一键制作应用镜像上架」的控制台功能，支持对于云机上已安装的应用一键制作Duapk镜像，提升客户操作体验和效率。	实例列表
		营销套件批量部署功能增加提示，提升仿真效果。	实例列表
	仿真授权管理	授权码换绑应用功能，支持将已绑定于某一应用的授权码变更绑定在其他应用上，方便客户快速更换仿真环境。	仿真授权管理
	商品类型管理	创建商品类型的逻辑优化，提升云手机速享版商品套餐配置的灵活性。	商品类型管理
	AI 内容生成	新增「AI 内容生成」页面，支持客户在平台自主制作数字人视频，生成品牌联名运营海报，内嵌留资导流通道，大幅降低素材生产门槛。	AI 内容生成
	API功能	新增「一键Duapk制作应用镜像上架」接口，支持对于云机上已安装的应用一键制作Duapk镜像，提升客户操作体验和效率。	制作Duapk镜像-高级模式
	帮助文档	H5 SDK文档更新。	SDK文档

2026年1月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	新增实例SN、物理机规格的展示和筛选，协助客户实时查询对应信息。	实例列表
	实例池管理	实例池列表新增设备数据统计，支持客户实时查询不同实例池的实例总数。	实例池管理
	API功能	新增实例池分页查询接口，支持客户通过接口实时查询实例池信息和设备库存数量。	客户实例汇总数据查询
		新增实例池设备变更通知接口，协助客户及时了解实例池设备数量变化。	实例池数量变更通知
		实例状态变更接口新增实例故障状态变更通知，协助客户实时了解和及时处理故障设备。	实例状态变更通知接口
		新增ROOT任务结果通知回调接口，协助客户及时了解ROOT任务执行结果。	实例ROOT/UNROOT结果通知
云手机速享版/SDK	云手机速享版	H5/安卓客户端新增切换云手机功能，可以快速切换至上一台/下一台云手机，方便用户使用。	/
		安卓客户端单笔订单批量购买云手机逻辑优化，提升用户使用体验。	/
	Android SDK	解决摄像头推流时存在黑屏的场景。	/

2025年12月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	支持非独立挂载的存储隔离设备自助调整开数，提升客户运营效率。	实例列表
	AI Agent	提供AI Agent入口，使用VLA大模型驱动云手机，客户可以通过输入文字指令控制云机完成特定任务。	AI Agent入口
	Duapk上架	Duapk制作的应用镜像支持跨安卓系统使用，提升客户镜像制作效率。	/
	帮助文档	帮助文档-“最佳实践”目录下更新Duapk使用文档。	Duapk使用文档
	API功能	提供在线快照备份接口，备份云手机快照时，可以不重启云手机。	/
		新增存储隔离设备查询接口。	/
		新增物理机状态变更通知接口。	物理机状态变更通知
		新增分页查询营销套件授权信息接口。	分页查询营销套件授权信息
		新增Duapk批量还原安装应用的接口。	实例批量安装Duapk应用镜像
云手机速享版/SDK	H5 SDK	按键位置异常bug修复。	/
		增加批量控制预览并且支持1000个云手机批量控制。	/
		修复推流绿屏问题。	/
		提升屏幕下拉操作体验。	/

2025年11月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	aPaaS开放平台新增"释放当前预览会话"功能。	实例列表
	脚本自动化	aPaaS开放平台支持脚本自动化功能，支持脚本管理、任务管理，实现多设备的规模化、自动化操作能力。	自动化
	API功能	新增壁纸设置接口。	/
		优化任务结果查询的接口，增加任务类型、任务创建时间和任务执行完成时间字段。	任务结果查询
		新增获取任务类型列表的接口。	获取任务类型列表
		新增分页查询任务信息的接口。	分页查询任务信息
		新增云机应用清理缓存的接口。	应用清理缓存
云手机速享版/SDK	云手机速享版	PC客户端优化批量卸载功能，支持搜索或选择应用进行批量卸载，让用户操作更直观高效。	/
		PC客户端「联系客服」入口位置调整优化。	/
	Android SDK	支持推流平均帧率、卡顿率、平均时延等关键指标的埋点上报。	/
	H5 SDK	优化摄像头推流的编码逻辑，解决摄像头推流时存在黑屏的场景。	/

2025年10月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	实例列表支持查询并导出配置网络信息，协助客户快速排查问题。	实例列表
	云手机盒子管理（仅限开通盒子权限客户可见）	业务中心新增盒子用户管理，管理PC客户端用户账号密码。	盒子用户管理
		物理机管理新增盒子物理机列表，展示所有已购盒子设备信息。	盒子物理机列表
		实例管理新增盒子实例列表，展示盒子创建的所有实例信息。	盒子实例列表
		任务管理新增盒子工作流列表，展示盒子实例任务详情。	盒子工作流列表
	API功能	实例备份接口支持指定文件目录，节约存储成本。	实例备份
		支持查看物理机历史资源占用情况，辅助客户追溯运行状态，快速定位问题。	获取实例性能数据
		新增实例状态变更通知接口，实时获取实例状态。	实例状态变更通知
		截图接口支持bmp图片格式，提升截图的清晰度。	实例截图
云手机速享版/SDK	云手机速享版	新增分段计价逻辑，可以自定义分段区间及对应价格，提升客户商业化计价的灵活性。	套餐管理
		新增拓展服务工具插件，包含超级管理权限、隐私守护、进程守护等9项功能，降低接入开发门槛。	/
	Android SDK	云游戏快进快出场景体验优化，推流服务新增回调，确保客户多次退出进入的流畅体验。	/
	H5 SDK	云游戏快进快出场景体验优化，推流服务新增回调，确保客户多次退出进入的流畅体验。	/

2025年9月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	在实例列表实现营销套件部署的界面化操作，快速筛选未部署营销套件，批量一键部署，部署步骤从6步降低为2步，大幅提高客户运营效率。	实例列表
	标签管理	标签名称提升字符上限，提升客户自定义标签体验。	标签管理
	API功能	新增查询实例出口IP接口。	获取实例出口IP
云手机速享版/SDK	云手机速享版	PC客户端连接弹窗优化，提升客户群控体验。	/
		「激活码」和「激活码批次」菜单位置调至「商品管理」的二级菜单，提升客户操作效率。	激活码批次
		「实例等级管理」页面支持自定义实例等级。	实例等级管理
		PC客户端、安卓端、H5端综合优化重启云手机任务时的用户体验。	/
	Android SDK	优化云机输入法的重置逻辑。	/
	H5 SDK	优化云机输入法的重置逻辑。	/
		优化H5端推流速度。	/
		解决canvas与video渲染器之间的“副作用”。	/
		Web API更新，优化用户体验。	/
		新增WebTransport方案。	/

2025年8月

平台	变更模块	变更详情	跳转链接
官网控制台	业务管理	兼容官网子用户账号管理主用户已购资源，适配客户多部门协同下的管理员账号管理和资源管理，提升灵活性与便捷性。	业务管理
aPaaS	带宽统计	“高级查询”功能支持查询子商户/实例池/指定云手机范围在指定时间段内的带宽用量明细，并可自定义汇总计算方式，支持数据导出，辅助客户更高效进行内部拆分带宽用量。	带宽统计
	首页	平台首页应用存储用量统计看板支持实时监控DuAPK存储用量，速享版客户精准管控DuFS付费资源。	首页
	应用市场	应用市场开关灵活配置，预置应用支持添加至主屏且可自定义排序，应用更新提醒，满足客户触达下游用户更高效推广业务应用的需求。	应用市场
	实例等级管理	新增实例等级管理，支持自定义实例等级（含名称与ICON），简化配置流程。	实例等级管理
	实例列表	定制化任务支持上传本地脚本文件。	实例列表
		设备信息增加网段字段，并支持按网段搜索设备。	实例列表
	应用列表	应用安装时支持按“应用版本”筛选目标实例。	应用列表
	商户管理	平台可视化操作和API均支持批量创建商户/子商户。	商户管理
	账号管理	平台可视化操作和API均支持批量创建用户账号。	账号管理
	实例池管理	平台可视化操作和API均支持批量创建实例池。	实例池管理
云手机速享版/SDK	云手机速享版	新增查询云机内已安装全部应用的接口。	分页查询应用列表
		速享版技术文档上线。	OEM技术文档
	H5 SDK	优化H5端推流速度，降低海外代理在网页端控制设备的延时，加速推流。	/

2025年7月

平台	变更模块	变更详情	跳转链接
aPaaS	实例列表	直连操作支持将云机文本复制粘贴至本地。	实例列表
		优化云手机实例预览展示布局，每行能够预览完整的云手机画面。	实例列表
	应用管理	支持通过内置应用接口将应用进行内置安装并可添加到云手机主屏页面，且支持自定义排序。	桌面推荐图标刷新
云手机速享版/SDK	鸿蒙SDK	BUG修复：摄像头前后置频繁切换偶现黑屏无画面、音频未释放掉流导致其他所有应用都启流失败问题修复。	/
	H5 SDK	BUG修复：鼠标滑动偶现上下漂移。	/
	Android SDK	云机通过app通话时，支持自动过滤本机的外放声音。	/
	云手机速享版	支持分应用配置是否开启应用强制更新，客户自主选择是否提示应用更新，业务运营更加灵活。	桌面配置更新
		PC客户端云手机预览支持视频流预览模式，画面加载速度更快，效率更高。	/
		自定义PC客户端时，新增支持任务栏图标自定义。	/

2025年6月

平台	变更模块	变更详情	跳转链接
官网控制台	概览	新增国内/海外云手机购买、SDK下载、API中心、售前/售后服务等快捷入口，并加强新手引导、支持快速购买等新客体验。	/
	实例管理	新增实例预览列表，提升可视化操作体验。	/
		新增更新镜像、实例限速、执行脚本、内存限额、开启维护、实例调度等丰富实例操作，精准跳转至项目管理平台操作提升平台使用的连贯性。	/
aPaaS	资源监控	新增存储资源用量实时监控及阈值告警，协助提升业务资源利用率。	/
		新增带宽流量资源实时监控及阈值告警，实时掌握业务资源使用情况。	带宽监控
	快照管理	新增快照多对多批量挂载接口，大幅提升业务对接效率和接口灵活性。	快照批量挂载
	实例管理	新增实例间数据对拷功能，无需对象存储也可实现无损换机。	实例对拷
		批量控制新增实时推流模式，降低时延、提升批量操作体验。	/
		新增实时获取实例内通知栏消息，提升私域运营效率，避免商机流失。	消息通知
	镜像管理	提升公共镜像和自定义镜像能力，支持实例/物理机更新镜像，实现自助刷机。	镜像管理
云手机速享版/SDK	鸿蒙SDK	全新发布鸿蒙SDK。	/
	DUAPK管理	DUAPK新产品发布，实现更高效热更新。	/
	SDK配置管理	桌面应用推荐实现按实例等级配置，并支持按套餐实例规格配置推荐应用榜单，辅助业务精细化运营。	/

产品介绍

什么是云手机实例

云手机实例

百度云手机基于自主知识产权的磐玉蜂巢服务器和容器虚拟化软件技术，通过在云端虚拟的原生安卓系统实例，为客户提供高性价比、弹性灵活、稳定可靠的IaaS+PaaS服务。百度云手机产品已广泛应用于文娱、教育、电商、金融、新零售、餐饮、旅游等行业，基于端云同构、原生应用兼容、信息数据安全、规格灵活调整、资源弹性伸缩等技术优势，为客户提供云端原生安卓工作环境，助力客户业务的自动化和智能化实现。

产品优势

高性价比

百度磐玉蜂巢服务器支持多款主流芯片，高密设计架构可支持单台服务器虚拟上百台云手机，每台云手机均具有原生操作系统，更高效资源利用、更低运营成本

效能提升

百度云手机提供功能强大的管理后台，支持批量运维，统一管控，具有丰富SDK接口，可实现业务灵活配置、资源调度，提升运营效能

业务安全

百度云手机运行的客户数据的计算和存储在云端，数据不落地，业务运营环境统一配置管控，保护业务数据安全

应用场景

面向游戏行业

- 云手机可实现游戏免下载试玩、云手游等，赋予游戏全新的互动体验方式。

面向政企、金融行业

- 为企业提供更加安全高效的移动办公解决方案，公私数据分离，信息安全更有保障。

面向新媒体行业

- 面向私域营销场景，为企业提供云手机私域营销算力与SCRM系统

落地场景

云游戏

场景介绍：

面向云游戏和云试玩场景，目标客户包括游戏开发商、运营商、视频网站等泛游戏类客户。

场景特点：

主打多(支持游戏大作多)、快(即点即玩)、好(4K高清和高达60FPS帧率的极致游戏体验)、省(订阅模式)等优势。

云直播

场景介绍：

面向云直播和一播多场景，目标客户包括运营商、直播基地等。

场景特点：

主打直播监播、离线文件转直播、虚拟偶像直播、无人直播、一机多播等直播带货新玩法，节省运营成本（支持24小时在线，节省主播数量及终端成本，自动化处理直播素材）。

安全办公云手机

场景介绍：

面向政企办公场景，目标客户包括公检法司、央企等业务安全要求高的客户。

场景特点：

主打办公数据保密（数据可用不可取，端外一律不可见；云端调取数据，终端显示信息；统一部署发布，分级设置权限）、工作和生活终端分离。

实例规格

实例规格

百度云手机的计费模式为**预付费和按量后付费**模式。具体规则如下：

- 预付费需要提前一次性支付几个月或者几年的费用。
- 按量后付费提交试用申请，然后提交工单联系客服。
- 购买实例前需保证账户无欠款。

实例规格详情如下：

以下为单片云手机物理机规格，单片物理机可自定义虚拟多个云手机实例：

商品	说明
云手机 G392 旗舰型	基于高通845，配置8核CPU8G内存128G存储，适用于云手游场景
云手机 R296 旗舰型	基于RK3588，配置8核CPU16G内存256G存储，适用于云游戏、云托管、云营销等场景
云手机 G272 旗舰型	基于高通8550，配置8核CPU24G内存512G存储，适用于云游戏、数字人直播、AI大模型实训等场景

通用部分

产品术语

- 磐玉蜂巢服务器

  磐玉蜂巢服务器是百度自主研发的基于ARM架构打造的具有仿生算力、高能效比、高性价比、高密设计的云边协同算力矩阵。为企业级数据中心和边缘计算场景提供高密服务器，广泛应用于安卓云算力和通用算力场景。

- 云手机Android系统

  基于AOSP（Android Open-Source Project）开源源码进行定制研发，可在ARM板卡运行，并且具备虚拟化能力，同一物理设备可运行多个Android实例，云手机Android系统与真机一致，并具备高度定制能力，满足多类业务需求。

- PaaS服务

  提供在线手机服务的能力，使虚拟机在安全、可视、可监管的环境下运行，使用视频传输的方式将系统画面传输至用户终端产品上，为用户提供传统手机的云端操作模式；同时通过管理平台以租户的形式对内对外提供服务，各租户由各自的管理员进行监控与授权。

- SaaS服务

  在云游戏、云直播等场景，百度致力于与合作伙伴共同建设行业解决方案，促进和推动生态发展。

- 全平台接入SDK

  提供Android SDK和H5 SDK接入方式，为开发者接入云手机、云游戏产品提供全平台接入方式。

什么是云手机（速享版）

1 产品概述

1.1 服务介绍

云手机（速享版）是面向以运营能力为主的客户，提供全套云手机端能力+运营管理平台的产品，旨在让客户用较少的技术投入，快速搭建自己的云手机产品，轻松打造自主云手机品牌。

1.2 与云手机服务的差异

- **客户端能力：**云手机（速享版）产品提供PC客户端、安卓APP、H5端及管理平台，客户端提供完成的UI交互功能，可满足C端用户各场景需要。
- **SDK能力：**针对自有业务的商户，云手机（速享版）产品提供SDK，支持商户在自己的APP中集成SDK，便捷集成，快速入驻商户开放平台。
- **运营能力：**商户可根据业务规划，自主定义商品套餐在自主品牌的客户端APP中进行销售获取盈利。同时，云手机（速享版）产品提供商户开放平台为商户提供云手机平台管理服务，支持商户自主管理用户、云手机、商品、订单、权限等操作，商户可根据业务需要发起运营活动。
- **应用市场能力：**云手机（速享版）产品镜像中提供应用市场，商户可内置应用，支持应用一键安装。

产品优势

- 功能丰富

  提供完整客户端、SDK等接入方式可选，已经支持常用云手机平台所需要的各项功能，对齐一线云手机平台。

- 快速接入

  无需漫长开发及测试周期，成品APP及管理平台支持快速商用。

- 应用兼容

  兼容市面常见应用，用户体验与主流平台保持一致。

- 极低投入

  无需研发人员及测试人员投入，大幅降低上线成本。

应用场景

云游戏

免安装，即点即玩，减少游戏软件与手机硬件的兼容性问题，低配手机玩高配游戏。

云托管

支持设备24h在线，设备稳定，数据安全，一键托管，省心无忧。

云手机（速享版）接入指南

百度云手机（速享版）是百度面向客户打造的从IaaS到SaaS层的一站式云手机解决方案，平台提供全流程标准化工具与端到端技术支持，无需复杂开发能力，即可快速搭建独立自有云手机品牌，高效完成业务落地与商业化运营。核心接入流程如下：

1 制作专属客户端

平台支持PC、安卓、H5三端客户端的标准化制作，全程采用可视化操作，无需编写代码。上传符合规范的自定义品牌素材可打造专属视觉风格客户端。其中H5客户端无需打包，生成专属访问链接即可直接投入使用；PC与安卓客户端支持一键打包，完成后即可下载安装包投入使用。生成的客户端功能丰富完善，支持批量操作、同步操作等核心能力，适配规模化业务运营场景。

2 后台服务配置

平台提供标准化的第三方服务接入指引，覆盖业务运营核心后台模块，助力快速完成基础服务搭建。支付服务支持支付宝、微信两大主流收款渠道，针对不同客户端类型提供清晰的接入步骤与参数配置说明；短信服务兼容多家主流短信运营商，支持一键配置接入；合规协议方面，平台提供标准化配置入口，支持上传授权协议、隐私政策、用户使用协议等各类合规文件，完成上线前的合规准备工作。

3 套餐体系搭建

依托平台成熟的标准化商品套餐体系，可快速搭建适配自身业务的商业化售卖体系。体系涵盖实例等级、商品、套餐三个层级，可根据云手机性能配置自定义实例等级标识，创建标准化售卖商品单元，并进一步设置不同套餐的定价、使用时长、展示顺序等规则，灵活适配零售、批量采购等多种售卖场景，快速实现业务变现。

4 团队权限分配

平台内置完善的角色与权限管理系统，支持精细化的团队协作管控，支持根据团队成员工作分工，创建不同的角色身份，并为各角色分配对应的操作权限，覆盖实例管理、商品管理、账号管理、运营管理等核心模块，既保障团队高效协同，又能实现业务数据与操作权限的安全隔离。

5 下游设备运维

平台提供两种标准化的下游用户实例交付方式，满足不同业务场景的设备分发需求。支持直接将可用云手机实例绑定至指定用户账号，实现一对一精准交付；也可批量生成不同配置的激活码，通过线上线下渠道进行分发，用户凭激活码即可自主激活使用云手机服务。

百度云手机（速享版）通过全流程标准化的产品能力与端到端技术支持，大幅降低云手机业务准入门槛与接入周期，助力客户快速抢占市场，实现自有云手机业务的规模化发展。

快速入门

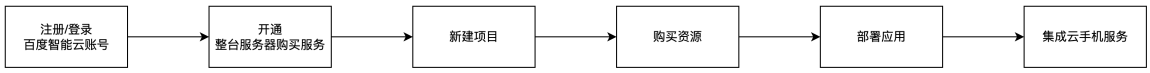
快速入门流程

概述

百度云手机是基于自研虚拟化技术，具有安卓实例及手机功能的云服务器。您可以远程实时控制云手机，实现安卓App的云端运行。基于云手机，可高效搭建应用，如云营销、云游戏、云直播和安全办公云手机等场景。

云手机接入流程

云手机接入流程如下图所示：



1.注册及实名认证：注册百度智能云账号，并完成企业实名认证。
2.开通云手机服务：申请免费试用国内/海外云手机，然后进入管理控制台，再进入默认项目测试试用服务。
2.购买云手机实例：进入管理控制台的国内业务/海外业务，新建项目，进入项目的物理机管理列表，通过购买入口下单国内/海外云手机。
3.使用云手机实例：进入管理控制台的项目，通过实例管理列表进行实例操作，点击“更多管理功能”进行应用管理、镜像管理、SDK下载等更多操作。

如需要批量购买云手机，或需要提供定制化服务，可提交工单联系我们。

注册账号

首先您需要注册/登录百度智能云账号，并完成企业实名认证。

开通云手机服务

1.申请免费试用：打开云手机的产品介绍页，点击“申请免费试用”，提交您的需求，您的需求会在5分钟内完成审核，并且百度智能云商务人员会在24小时内与您建立联系。如未收到，可提交工单联系我们。
2.试用云手机：访问云手机管理控制台，进入业务管理，进入已开通的默认POC项目，查看并试用免费云手机资源和服务。

购买云手机实例

1.新建项目：访问云手机管理控制台，左侧导航栏选择“业务管理”，进入国内业务/海外业务，点击“创建项目”，提交项目基本信息。
2.下单购买：进入刚创建的项目，左侧导航栏选择“物理机管理”，通过物理机列表页面的“购买”入口，进入下单页。下单页选择云手机规格、付费方式、创建实例个数、下单数量、时长等信息，然后提交订单支付即可。

（温馨提醒：因云手机配套带宽为后付费商品，下单前请确认账号最少有500元余额。如批量下单云手机时发现库存不足，可提交工单联系我们。）

使用云手机实例

1.部署应用：购买资源后，回到相应项目，左侧导航栏选择“更多管理功能”，进入业务管理平台，上传应用文件，完成应用安装部署，为后续集成客户端 SDK 时，获取指定实例资源的推流信息做准备。
2.集成云手机服务：如需接入SDK（Android、Web/H5）、OPEN API，开发定制化的客户端，集成云手机服务，请进入相应项目，左侧导航栏选择“更多管理功能”，进入业务管理平台，平台右上角“帮助文档”自行下载相应文件。

如您在云手机服务接入过程中遇到问题，欢迎提交工单联系我们。

购买云手机实例

概述

本文将向您介绍如何通过控制台购买云手机实例。

备注：如您对以下操作有疑问，请提交工单联系我们。

前提条件

- **注册百度智能云账号并实名认证**

  云手机仅限于为企业用户提供服务，使用百度智能云云手机BAC前，需要拥有一个百度智能云账号并完成企业实名认证，具体操作如下：

  1. 注册并登录百度智能云平台，请参考 注册 和 登录。
  2. 完成实名认证，操作细节请参考 实名认证，完成企业实名认证后才可购买百度智能云云手机BAC。

购买云手机

百度云手机支持物理机灵活购买和整台服务器购买，整台服务器购买可提交工单联系我们。 购买流程图：

> **申请接入百度云手机-->接入审核-->审核通过-->购买实例**

1. 新用户初次登录云手机 BAC 管理控制台，进入云手机BAC控制台。
2. 自动弹出“欢迎接入百度云手机”的申请表单，同时我们为新用户提供免费试用的限时限量福利，表单填写试用需求。
3. 填写申请表单，点击“确认”，进入到审核阶段，提交后预计10分钟内完成审核。
4. 审核通过后，回到云手机 BAC 管理控制台，点击**业务管理 > 国内业务/海外业务 > 创建项目 > 进入项目 > 物理机管理 > 物理机列表**，通过“购买”进入下单界面。
5. 根据需求选择所需要的配置后，点击**确定**，进入“确认订单”页面。

6. 确认选择的云手机套餐和价格后，点击**确定**，进入支付环节。

7. 点击**确认支付**，完成支付。支付成功后，系统在后台进行云手机实例的创建。

8. 购买成功后，回到**管理控制台**，进入“实例列表”界面，可以查看到对应购买实例列表。

> 温馨提示：目前云手机BAC产品支持物理机购买，单片物理机支持灵活多开。即下单数量为物理机，配置创建实例数量为云手机多开实例数。

计费模式

百度云手机包括国内云手机和海外云手机，云手机服务费用包括云手机资源服务费和带宽费用。

云手机资源支持预付费和后付费两种计费方式；带宽支持按带宽月95峰值、按带宽月峰值、按带宽日峰值和按实际流量四种后付费计费方式。可根据业务需要，选择不同的计费方式订购云手机服务。

云手机资源

- **预付费模式**

定义：先付费后使用服务。您可通过预充值账户余额直接下订单购买云手机，或通过百度智能云支持的支付方式直接下订单购买，下单时扣费。足额付费且成功订购后，百度智能云才开始为您提供服务。

预付费方式式支持包年包月预留资源，享有更优惠的价格，适合中长期稳定业务需求。

- **后付费模式**

定义：先使用服务后付费。您购买百度智能云按订购量计费的服务。您可先开通和使用云手机服务，百度智能云按实际使用时长计算费用、出具账单，并按账单金额扣减您的百度智能云账户余额。

后付费模式支持以天为单位进行计费，适合有短期弹性云资源需求的业务场景。

不同计费模式的对比说明如下：

计费模式	计费周期	适用场景	时间限制
包年包月	包年/包月预付费	适合中长期稳定业务需求	单次购买时间1个月起购
按量计费（按天）	以天为单位后付费	适合短期弹性云资源需求	无起购时间限制

带宽流量

带宽支持按带宽上限计费和按实际流量计费两种模式，其中，带宽上限计费模式支持多种计费方式。

不同计费方式的对比说明如下：

计费模式	计费方式	适用场景	费用计算
带宽	按带宽月95峰值	适用于流量较大或较稳定的场景	按当前业务实际使用的公网带宽月95峰值计费，按月结算
	按带宽月峰值	适用于流量较大或较稳定的场景	按当前业务实际使用的公网带宽月峰值计费，按月结算
	按带宽日峰值	适用于流量波动较大的场景	按当前业务实际使用的公网带宽日峰值计费，按天结算，隔两日出账 例：2024年5月1号的日峰值带宽用量，将统计在2024年5月2日的账单周期，于2024年5月3号出具账单再进行扣费
流量	按实际流量	适用于流量较小的场景	按当前业务实际使用的公网实际流量计费，按日结算 例：2024年5月1号的实际流量用量，将统计在2024年5月2日的账单周期，于2024年5月3号出具账单再进行扣费

> 温馨提示：同个机房，带宽的计费方式只可开通一种，具体情况可联系百度智能云沟通，确认后再下单。

起购限制

云手机服务开启后，会同步开通实时音视频传输功能。在使用过程中，会产生相应的带宽用量，并按周期从您的账户余额中扣除相关费用。为避免由于账户余额不足导致服务中断，申请购买云手机服务时，您的账户总资产必须达到500元以上。如果账户余额低于此限额，将无法完成购买。

续费说明

在您的百度智能云余额不足以支付您下一个服务周期的服务费、当前余额不足以支持当前服务，或您此时的服务周期即将到期时，我们会通过站内信/邮件/短信的方式对您进行提醒，避免影响您的服务体验。

提醒类型	提醒方式	提醒时间及频次	操作
预付费延期续费	站内信/邮件/短信	提前7/3/1天和当天，每次发送一次提醒	若您开通了自动延期续费，请前往控制台确认延期时间。我们在延期续费节点进行自动延期续费时，默认优先使用代金券，您可在支付时进行取消优先使用代金券的延期续费方式。
后付费余额不足	站内信/邮件/短信	提前3/2/1天和当天，每次发送一次提醒	在您在提醒周期内没有及时进行充值，我们会终止您的服务，对您的数据及信息进行统一释放。

到期回收规则

当您购买了包年包月的资源，到期后状态变化如下：

1. 到期7天内，到期资源进入赎回期，到期资源会变为过期关停状态，云手机实例无法继续使用，平台会继续保留您的云手机数据，期间续费成功后即可恢复正常使用。
2. 到期超过7天，云手机实例将会被回收释放，资源回收后相关数据也将被清空。

退订说明

如果您在购买后有任何不满意，均可通过提交工单联系我们。

1. 云手机预付费资源，如遇到产品不可用问题（如云手机连接闪退、云手机系统不兼容应用等），购买后7天内可退订，费用以系统实际计费天数为准来收取，超过7天不支持退订。退订申请通过后，您的云手机实例将会被立即回收释放，资源回收后相关数据也将被清空。
2. 云手机后付费资源，如需退订，请提前30天提交工单发送退订申请，如退订申请通过，则自工单发出30个自然日后该退订

申请生效，退订生效后，您的云手机实例将会被立即回收释放，资源回收后相关数据也将被清空。

请您在仔细了解并试用云手机后再进行购买。

在您退订申请通过后，预付费对应金额返回至您的百度智能云账户中。

> 温馨提示：云手机预付费资源，非产品不可用问题，不支持退订，百度智能云负责对本协议进行解释说明。如出现疑似异常/恶意退货，百度智能云有权拒绝您的退订申请

解释权说明

购买指南的最终解释权归百度智能云云手机所有。

购买营销套件服务

概述

本文将向您介绍如何通过控制台购买营销套件。

备注：如您对以下操作有疑问，请提交工单联系我们。

前提条件

1. 购买营销套件的前提为拥有未过期状态的云手机，未订购云手机无法单独购买营销套件。
2. 营销套件最短计费时长为一个月，因此建议您的云手机剩余有效期大于等于一个月。若云手机因过期导致回收，营销套件也无法提供服务。

购买营销套件

营销套件以预付费的方式购买。 购买流程图：

> **管理控制台-->购买云手机配套服务**

1. 登录云手机 BAC 管理控制台，点击业务管理-国内业务/海外业务，右上角“购买云手机配套服务”。
2. 根据需求选择所需要的营销套件数量和订购时长，点击下一步，确认订单，进入“确认订单”页面。
3. 确认选择的营销套件的数量、时长和价格后，点击提交订单，进入支付环节。
4. 点击确认支付，完成支付。支付成功后，系统在后台进行营销套件的创建。
5. 购买成功后，回到管理控制台，进入业务中心>仿真授权管理，跳转至项目管理平台可以查看到对应购买授权码列表，平台右上角帮助文档-云手机-最佳实践可查看营销套件部署方法。

计费模式

营销套件支持预付费的计费方式。

定义：先付费后使用服务。您可通过预充值账户余额直接下订单购买营销套件，或通过百度智能云支持的支付方式直接下订单购买，下单时扣费。足额付费且成功订购后，百度智能云才开始为您提供服务。

计费模式	计费周期	适用场景	时间限制
包年包月	包年/包月预付费	适合中长期稳定业务需求	单次购买时间1个月起购

续费说明

在您的百度智能云余额不足以支付您下一个服务周期的服务费、当前余额不足以支持当前服务，或您此时的服务周期即将到期

时，我们会通过站内信/邮件/短信的方式对您进行提醒，避免影响您的服务体验。

提醒类型	提醒方式	提醒时间及频次	操作
预付费延期续费	站内信/邮件/短信	提前7/3/1天和当天，每次发送一次提醒	若您开通了自动延期续费，请前往控制台确认延期时间。我们在延期续费节点进行自动延期续费时，默认优先使用代金券，您可在支付时进行取消优先使用代金券的延期续费方式。

到期回收规则

当您购买了包年包月的资源，到期后状态变化如下：

1. 到期7天内，到期资源进入赎回期，未绑定的营销套件不能再进行绑定，已绑定的营销套件相关功能将不可使用，期间续费成功后即可恢复正常使用。
2. 到期后，营销套件相关功能将不可使用。

退款退订说明

如果您在购买后有任何不满意，均可通过提交工单联系我们。

营销套件不支持线上退订，需通过提交工单申请退订。开通营销套件服务后，单批服务使用时长不得少于30个自然日，不足30日按30日计费。 若使用超30日要求提前终止订单，按实际使用时长计费。

> 温馨提示：云手机配套服务预付费资源，非产品不可用问题，不支持退订，百度智能云负责对本协议进行解释说明。如出现疑似异常/恶意退货，百度智能云有权拒绝您的退订申请

解释权说明

购买指南的最终解释权归百度智能云云手机所有。

产品定价

计费概述

云手机服务开启后，会同步开通实时音视频传输功能。在使用过程中，会产生相应的带宽用量。云手机服务费用 = 云手机资源服务费 + 带宽费用。

计费规则

计费项	计费方式	配置	计费周期	说明
云手机资源	预付费	• 8vCPU | 8GB内存 | 128GB存储 • 8vCPU | 16GB内存 | 256GB存储	包月	• 先付费后使用的方式。下订单时，根据云手机数量、订购时长确定云手机服务费用，付费后资源生效可使用。 • 包月云手机资源服务费=单次订购实例数量*订购时长*包月单价（时长最小单位为1个月起）
	后付费	• 8vCPU | 8GB内存 | 128GB存储 • 8vCPU | 16GB内存 | 256GB存储	按日	• 客户订购当天开始计费，按日峰值订购量计算。 • 日云手机资源服务费=日订购云手机峰值数量*云手机按日单价。 • 按日为结算周期，每日出具前一日服务账单并按账单金额扣费
带宽流量	后付费	按月95峰值	按月	• 在一个自然月内，每5分钟统计同一时刻入网带宽和出网带宽数据较大值，每日得到288个采集值，然后将所有的采集值进行降序排序，其中Top 5%的采集值去掉，剩余最高值为月95峰值带宽。 • 每月带宽费用=月带宽95峰值*月95峰值带宽单价 • 按月为结算周期，每月前4个工作日内出具上一自然月服务账单并按账单金额扣费
	后付费	按月峰值	按月	• 在一个自然月内，每5分钟统计同一时刻入网带宽和出网带宽数据较大值，每日得到288个采集值，然后将所有的采集值进行降序排序，取最高值计量。 • 每月带宽费用=月带宽峰值*月峰值带宽单价 • 按月为结算周期，每月前4个工作日内出具上一自然月服务账单并按账单金额扣费
	后付费	按日峰值	按日	• 在一个自然日内，每5分钟统计同一时刻入网带宽和出网带宽数据较大值，每日得到288个采集值，然后将所有的采集值进行降序排序，取最高值计量。 • 每日带宽费用=日带宽峰值*日峰值带宽单价。 • 按日为结算周期，每日出具前一日服务账单并按账单金额扣费
	后付费	按实际流量	按日	• 在一个自然日内，产生的所有流量之和。 • 每日流量费用=日实际流量*流量单价 • 按日为结算周期，每日出具前一日服务账单并按账单金额扣费

> 按月带宽计费的特殊说明:

1. 计费周期：按自然月计费
2. 结算规则：按照当月带宽费用乘以有效因子计费

- 有效天数：按月95峰值计费生效日开始到该月月末的天数。例如2021年11月5日为按月95峰值计费生效时间，则2021年11月的有效天数为26天。次月起，有效天数为整个自然月的天数。
- 有效因子：以一个自然月的天数为分母、该自然月内的有效天数为分子计算得到的小数。假设2021年11月的有效天数为26天，则有效因子为26/30=0.8666666667。

服务关停

预付费到期关停

- 预付费实例到期日前，可操作续费，成功续费即能继续正常使用。
- 未及时操作续费的，预付费实例到期日后超过7天，回收云手机实例资源，资源回收后相关数据将被清空。

账号欠费关停

- 账户可用余额不足或账户可用余额小于0被扣为负值即为欠费。
- 账号欠费后7天内补缴所有欠费账单的，可正常使用云手机服务。欠费后7天内仍未能补缴所有欠费账单的，将关停云手机所有服务（包括实例和带宽），关停后云手机服务将无法使用，充值可恢复正常。欠费7天后回收云手机实例资源和带宽资源。
    - 回收云手机实例资源后，实例列表将不可见相关实例。
    - 回收带宽资源后，实例将无法进行公网访问。

退订说明

- 云手机预付费资源和后付费资源的具体退订规则请参见[购买云手机实例]的退订规则说明。
- 以单个百度智能云账户维度发起云手机预付费实例退订申请，费用以系统实际计费天数为准来收取。

- 在您申请退订时，若相应产品/服务在购买时参与“活动”或使用“代金券”，优先适用当时活动或代金券相关规则。
- 退订申请通过后，相关退款金额返回至您的百度智能云账户中。

如百度智能云认为您的退订退款申请为疑似异常/恶意退订，百度智能云有权要求您配合提供合理理由以审核您的退订退款申请或拒绝您的退订退款申请。

关于价格咨询，欢迎提交工单联系我们。

技术文档

云手机Android SDK

百度云手机Android-SDK接入说明

1.综述

云手机：用户在运营平台购买实例后，通过接入SDK，指定进入云手机模式。

2.1 接入准备

2.1.2环境要求

- minSdkVersion 21（android 5.0）及以上
- GLES 2.0及以上
- gradle版本6.1.1
- sdk只支持arm架构android环境（支持32及64位系统）

2.2 SDK集成

2.2.1 aar集成方法

将sdk（.aar文件）拷贝到项目libs(**没有libs目录手动创建一个**)目录下，并在项目模块的build.gradle中配置相应参数。

```
android {
  defaultConfig {
     ndk {
        abiFilters "arm64-v8a","armeabi-v7a"
     }
  }
  repositories {
     //libs 目录
     flatDir {
        dirs "libs"
     }
  }
  dependencies {
     implementation fileTree(include: ['*.aar'], dir: 'libs')
     ...
  }
}
```

2.2.2 sdk所需权限

```
 <uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<!---------- 下面两项按需添加 ---------->
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

2.3 混淆规则

集成sdk后如需混淆，需要在相应的proguard-rules.pro文件中添加如下混淆规则

```
 -keep class com.mci.** { *; }
-keep class com.baidu.armvm.mciwebrtc.** { *; }
```

2.4 布局组件

在布局文件中添加**SdkView**组件

```
 <com.baidu.armvm.api.SdkView
      android:id="@+id/sdk_view"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_centerInParent="true"
      android:gravity="center"/>
```

2.5 Activity中接入SDK

2.5.1 接入流程图

连接实例流程图

断开连接实例流程图

2.5.2 预加载，初始化UI

接口：preLoad

示例代码：

```
public class DemoApplication extends Application {
    private static final String TAG = "DemoApplication";
    public static boolean sdkPreLoadSuccess = false;

    @Override
    public void onCreate() {
        super.onCreate();

        // 1、initialization environment
        HashMap<String, Object> params = new HashMap<>();
        params.put("preLoadListener",
                (PreLoadListener) (code, msg) -> {
                    Log.d(TAG, "preload so code:" + code + ", msg:" + msg);
                    if (code == 0) {
                        sdkPreLoadSuccess = true;
                    }
                });
        BgsSdk.preLoad(this, params);
    }
}

public class VideoPlayActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // 推荐设置Activity界面为全屏沉浸式效果
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);

        if (!DemoApplication.sdkPreLoadSuccess) {
            createErrorDialog(this, 100001, "PreLoadFail",
                    (dialog, which) -> VideoPlayActivity.this.finish());
            return;
        }

        setContentView(R.layout.activity_video_play);

        mSdkView = findViewById(R.id.sdk_view);
    }
}
```

2.5.3 创建sdk对象、初始化实例播放参数

接口：initPhone

示例代码：

```
 bgsSdk = new BgsSdk(VideoPlayActivity.this);

HashMap<String, Object> params = new HashMap<>();
params.put("sdkView", mSdkView);
params.put("uuid", mUuid);
params.put("width", 720);
params.put("height", 1280);
params.put("bitrate", 8192);
params.put("fps", 30);
params.put("sdkCallback", myPlaySdkCallback);
// 用SDK采集视频
params.put("useSdkCollectVideo", true);
// 用SDK采集音频
params.put("useSdkCollectAudio", true);

bgsSdk.initPhone(params);
```

2.5.4 获取ServerToken

通过访问接口server-token去服务端获取serverToken。

2.5.5 连接实例

接口：startPhone

示例代码：

```
 if (bgsSdk != null) {
   bgsSdk.startPhone(serverToken);
}
```

2.5.6 断开实例

接口：stopPhone

示例代码：

```
 if (bgsSdk != null) {
   bgsSdk.stopPhone();
}
```

2.5.7 解绑实例

在SDK回调onStopped后，可视情况通过业务后台调用openApi接口disconnect去服务端解绑实例。

2.5.8 完整示例

```
public class DemoApplication extends Application {
  private static final String TAG = "DemoApplication";
  public static boolean sdkPreLoadSuccess = false;

  @Override
  public void onCreate() {
     super.onCreate();

     // 1、initialization environment
     HashMap<String, Object> params = new HashMap<>();
     params.put("preLoadListener",
           (PreLoadListener) (code, msg) -> {
              Log.d(TAG, "preload so code:" + code + ", msg:" + msg);
```

```
                Log.d(TAG, "preload sdk code: " + code + ", msg: " + msg);
                if (code == 0) {
                    sdkPreLoadSuccess = true;
                }
            });
        BgsSdk.preLoad(this, params);
    }
}

public class VideoPlayActivity extends AppCompatActivity {
    public static final int REQUEST_CAMERA_CODE = 5;
    public static final int REQUEST_MIC_CODE = 6;
    private static final String[] PERMISSION_CAMERA = new String[]{
            Manifest.permission.CAMERA,
    };
    private static final String[] PERMISSION_MIC = new String[]{
            Manifest.permission.RECORD_AUDIO,
    };
    // 需要启动的应用ID
    private static final int APP_ID = 6;

    private MyPlaySdkCallback myPlaySdkCallback;

    private static final int INIT_PHONE = 1;
    private static final int STOPED_PHONE = 2;

    private BgsSdk bgsSdk;

    private SdkView mSdkView; // video渲染用view
    private View mBackView;

    private String mUuid; // 用户唯一id，不能为空

    private HandlerThread mHandlerThread;
    private Handler mHandler;
    private String serverToken;
    private HandlerNetwork handlerNetwork;

    /**
     * 获取uuid，此方法在部分手机上（例如小米11 android 11）可能获取不到，
     * 商用时获取uuid谨慎使用，此处仅用于演示
     *
     * @param application
     * @return
     */
    public String getUUID(Application application) {
        String uuid = null;
        try {
            uuid = Settings.System.getString(application.getContentResolver(),
                    Settings.Secure.ANDROID_ID);
        } catch (Exception e) {
            e.printStackTrace();
        }

        if (TextUtils.isEmpty(uuid)) {
            uuid = "mciDemoDefaultUuid";
        }
        return uuid;
    }

    private void initPhone() {
        myPlaySdkCallback = new MyPlaySdkCallback(VideoPlayActivity.this);
```

```
    HashMap<String, Object> params = new HashMap<>();
    params.put("sdkView", mSdkView);
    params.put("uuid", mUuid);
    params.put("width", 720);
    params.put("height", 1280);
    params.put("bitrate", 8192);
    params.put("fps", 30);
    params.put("sdkCallback", myPlaySdkCallback);
    // 用SDK采集视频
    params.put("useSdkCollectVideo", true);
    // 用SDK采集音频
    params.put("useSdkCollectAudio", true);
    bgsSdk.initPhone(params);
}

private void getServerToken() {
    if (bgsSdk == null) {
        return;
    }

    try {
        handlerNetwork.setCallBack(new HandlerNetwork.CallBack() {
            @Override
            public void onSuccess(String result) {
                Gson gson = new Gson();
                Response response = gson.fromJson(result, Response.class);

                if (response == null || response.getData() == null
                        || response.getData().getSuccessList() == null
                        || response.getData().getSuccessList().size() < 1
                        || response.getData().getSuccessList().get(0) == null
                        || TextUtils.isEmpty(response.getData()
                        .getSuccessList().get(0).getServerToken())) {
                    onDisconnected(222001, "获取serverToken失败");
                    return;
                }
                serverToken = response.getData()
                        .getSuccessList().get(0).getServerToken();

                if (bgsSdk != null) {
                    bgsSdk.startPhone(serverToken);
                }
            }

            @Override
            public void onFail(String error) {
                onDisconnected(222001, "获取serverToken失败");
            }
        });

        String[] instanceCodes = new String[] {"VM010151051179"}; //实列列表
        String onlineTime = String.valueOf(60 * 60 * 24 *30); // 在线时长1个月
        handlerNetwork.getServerToken(mUuid, instanceCodes,
                onlineTime);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private void stopPhone() {
    if (mHandler != null) {
        mHandler.removeCallbacksAndMessages(null);
```

```
            mHandler = null;
        }

        if (mHandlerThread != null) {
            mHandlerThread.quitSafely();
            mHandlerThread = null;
        }
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // set  full screen
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);

        if (!DemoApplication.sdkPreLoadSuccess) {
            createErrorDialog(this, 100001, "PreLoadFail",
                    (dialog, which) -> VideoPlayActivity.this.finish());
            return;
        }

        setContentView(R.layout.activity_video_play);
        mSdkView = findViewById(R.id.sdk_view);
        mBackView = findViewById(R.id.menu_view);
        bgsSdk = new BgsSdk(VideoPlayActivity.this);
        mUuid = getUUID(getApplication());
        handlerNetwork = new HandlerNetwork();

        mHandlerThread = new HandlerThread("NetworkRequestThread");
        mHandlerThread.start();
        mHandler = new Handler(mHandlerThread.getLooper()) {
            @Override
            public void handleMessage(@NonNull Message msg) {
                super.handleMessage(msg);
                switch (msg.what) {
                    case INIT_PHONE:
                        initPhone();
                        break;
                    case STOPED_PHONE:
                        stopPhone();
                        break;
                    default:
                        break;
                }
            }
        };
        mHandler.sendEmptyMessage(INIT_PHONE);
    }

    @Override
    protected void onResume() {
        super.onResume();
        // resume game
        if (bgsSdk != null) {
            bgsSdk.resume();
        }
    }

    @Override
    protected void onStop() {
        super.onStop();
```

```
    super.onStop();
    // pause game, 应用彻底不可见时暂停云端推流
    if (bgsSdk != null) {
        bgsSdk.pause();
    }
}

@Override
public void onBackPressed() {
    if (mBackView != null) {
        mBackView.setVisibility(View.VISIBLE);
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    // stop game and release resources
    if (bgsSdk != null) {
        // stop game
        bgsSdk.stopPhone();
    }

    myPlaySdkCallback = null;
}

public void onDisconnected(int errorCode, String msg) {
    createErrorDialog(VideoPlayActivity.this, errorCode, msg,
            (dialog, which) -> VideoPlayActivity.this.finish());
}

public void onInitSuccess() {
    getServerToken();
}

public void onInitFail(int code, String msg) {
    createErrorDialog(VideoPlayActivity.this, code, msg,
            (dialog, which) -> VideoPlayActivity.this.finish());
}

public void onConnectSuccess() {
}

public void onStopped() {
    if (mHandler != null) {
        mHandler.sendEmptyMessage(STOPED_PHONE);
    }
}

public void onClick(View view) {
    int keyCode = -1;
    switch (view.getId()) {
        case R.id.menu_btn:
            keyCode = MCIKeyEvent.KEYCOED_MENU;
            break;
        case R.id.home_btn:
            keyCode = MCIKeyEvent.KEYCOED_HOME;
            break;
        case R.id.back_btn:
            keyCode = MCIKeyEvent.KEYCOED_BACK;
            break;
        case R.id.quit_btn:
            // 退出
```

```
            finish();
            break;
        default:
            break;
    }

    if (keyCode > 0 && bgsSdk != null) {
        bgsSdk.sendKeyEvent(MCIKeyEvent.ACTION_UNDEFINED, keyCode);
    }

    if (mBackView != null) {
        mBackView.setVisibility(View.GONE);
    }
}

private void createErrorDialog(Context context, int errCode, String errMsg,
                               DialogInterface.OnClickListener onClickListener) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            AlertDialog.Builder b = new AlertDialog.Builder(context);
            b.setTitle("返回错误提示");
            b.setMessage("CallBack返回了" + errCode + " ; errMsg : " + errMsg);
            b.setNegativeButton("确定", onClickListener);
            b.setPositiveButton("取消", onClickListener);
            AlertDialog dialog = b.create();
            dialog.setCanceledOnTouchOutside(false);
            dialog.show();

        }
    });
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                                       @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == REQUEST_CAMERA_CODE) {
        if (bgsSdk != null) {
            bgsSdk.openCamera();
        }
    } else if (requestCode == REQUEST_MIC_CODE) {
        if (bgsSdk != null) {
            bgsSdk.openMic();
        }
    }
}

public void onRequestPermission(String s) {
    if (Manifest.permission.RECORD_AUDIO.equals(s)) {
        checkPermission(VideoPlayActivity.this, PERMISSION_MIC, REQUEST_MIC_CODE);
    }
    if (Manifest.permission.CAMERA.equals(s)) {
        checkPermission(VideoPlayActivity.this, PERMISSION_CAMERA, REQUEST_CAMERA_CODE);
    }
}

public static boolean checkPermission(Activity activity, String[] permission, int requestCode) {
    if (isPermissionGranted(activity, permission)) {
        return true;
    } else {
        // 如果没有设置过权限许可，则弹出系统的授权窗口
```

```
            ActivityCompat.requestPermissions(activity, permission, requestCode);
            return false;
        }
    }

    // 每个权限是否已授权
    public static boolean isPermissionGranted(Activity activity, String[] permission) {
        if (Build.VERSION.SDK_INT >= 23) {
            for (int i = 0; i < permission.length; i++) {
                int checkPermission = ContextCompat.checkSelfPermission(activity, permission[i]);
                /***
                 * checkPermission返回两个值
                 * 有权限: PackageManager.PERMISSION_GRANTED
                 * 无权限: PackageManager.PERMISSION_DENIED
                 */
                if (checkPermission != PackageManager.PERMISSION_GRANTED) {
                    return false;
                }
            }
        }
        return true;
    }
}

public class MyPlaySdkCallback extends BgsSdkCallback {
    private static final String TAG = "MyPlaySdkCallback";
    private VideoPlayActivity videoPlayActivity;

    public MyPlaySdkCallback(VideoPlayActivity videoPlayActivity) {
        this.videoPlayActivity = videoPlayActivity;
    }
    @Override
    public void onInitSuccess() {
        if (videoPlayActivity != null) {
            videoPlayActivity.onInitSuccess();
        }
    }

    @Override
    public void onInitFail(int i, String s) {
        if (videoPlayActivity != null) {
            videoPlayActivity.onInitFail(i, s);
        }
    }

    @Override
    public void onConnectSuccess() {
        if (videoPlayActivity != null) {
            videoPlayActivity.onConnectSuccess();
        }
    }

    @Override
    public void onConnectFail(int i, String s) {
        if (videoPlayActivity != null) {
            videoPlayActivity.onDisconnected(i, s);
        }
    }

    @Override
    public void onStopped() {
        if (videoPlayActivity != null) {
            videoPlayActivity.onStopped();
```

```
            videoPlayActivity.onStopped();
        }
    }

    @Override
    public void onDisconnected(int i) {
        Log.d(TAG, "onDisconnected i = " + i);
        if (videoPlayActivity != null) {
            videoPlayActivity.onDisconnected(i, "onDisconnected");
        }
    }

    @Override
    public void onPlayInfo(String s) {
        Log.d(TAG, "onPlayInfo s = " + s);
    }

    @Override
    public void onRenderedFirstFrame(int i, int i1) {
        Log.d(TAG, "onRenderedFirstFrame i = " + i + ", i1 = " + i1);
    }

    @Override
    public void onVideoSizeChanged(int i, int i1) {
        Log.d(TAG, "onVideoSizeChanged i = " + i + ", i1 = " + i1);
    }

    @Override
    public void onRequestPermission(String s) {
        if (videoPlayActivity != null) {
            videoPlayActivity.onRequestPermission(s);
        }
    }
}
```

函数说明：

1. 视情况调用pause暂停实例推流；
2. 视情况调用resume恢复实例推流。

3 SDK高级功能

3.1 切换分辨率，比特率，帧率

4 SDK开发指南

sdk的api函数都封装到了com.mci.commonplaysdk.BgsSdk.java类中，具体函数描述如下

4.1 构造函数

- 函数原型：

```
public BgsSdk(Activity activity)
```

- 函数说明：BgsSdk构造函数。

4.2 类函数

4.2.1 preLoad

- **函数原型：**

```
public static void preLoad(Application application, HashMap<String, Object> params)
```

- **函数说明：**预加载SO，初始化SDK运行时环境。
- **参数说明：**

参数	类型	说明
application	Application	非空Application对象
params	HashMap<String, Object>	加载SO的参数，见下表

- **params参数说明：**

关键字	值的类型	必填项	说明
preLoadListener	PreLoadListener	非必填	默认值为null。（SO加载回调，详见PreLoadListener）
cloudLoadSo	boolean	非必填	默认值为false。（true：加载网络SO， false：加载本地SO）
soFile	String	非必填	默认值为null。（SDK 中加载so库文件地址，不设置使用SDK中默认so库）
logLevel	int	非必填	BsgSdk.LOG_DEFAULT。（日志级别，具体定义参考log等级）
writeLogToFile	boolean	非必填	默认值为true。（标记是否把SDK中日志保存到设备本地，true ：保存；false：不保存）

- **返回结果：**

通过**PreLoadListener**接口中的**void onLoad(int code, String msg)**回调SO加载结果。

code：0，表示加载成功。

code：非0，表示加载失败，msg变量中给出相应的异常信息。

4.3 成员函数

initPhone

特别说明：初始化操作要确保在SO加载成功后再进行。

- **函数原型：**

```
public int initPhone(HashMap<String, Object> params)
```

- **函数说明:**用户信息参数、云手机定制参数。
- **params参数说明：**

关键字	值的类型	必填项	说明
sdkView	SdkView	必填	SdkView对象，云手机显示的画布，需要布局到您的应用
uuid	String	必填	设置用户uuid，uuid是用来区分真实用户的唯一标识，由应用测来定义，请根据业务场景来决定
width	int	必填	推流视频分辨率宽，如：720（推流请求是视频是竖屏width < height）
height	int	必填	推流视频分辨率高，如：1280
bitrate	int	必填	推流视频码率，单位kBps，如：4092

fps	int	必填	推流视频帧率，如：30
instanceCode	string	非必填	设备编号
sdkCallback	BgsSdkCallback	非必填	默认值为null。(SDK回调接口，详见BgsSdkCallback)
useSWDecode	boolean	非必填	默认值为false。（Video解码方式，true：软解；false：硬解）
noVideoDataTimeout	int	非必填	默认值为30。（单位s，设置首次启动视频推流超时时间，若超过设定超时时间还没有视频流返回，会报10006错误码，并断开与云端连接。）
gameScreenRotate	boolean	非必填	默认值为false。（控制Sdk是否允许画面依据手机重力方向做180度旋转。true，开启此功能；false，关闭此动能）
autoSwitchDecodeMode	boolean	非必填	默认值为false。（控制Sdk硬解异常后自动切换到软解。 false：不切换； true：切换）
forcePortrait	boolean	非必填	默认值为false。（设置强制横屏显示，设置后，竖屏应用会被强制横屏显示。针对电视、盒子等大屏设备，普通手机无需设置。）
defaultRotation	boolean	非必填	默认值为false。（设置应用默认屏幕方向，false：竖屏， true：横屏）
useSdkCollectVideo	boolean	非必填	默认值为false。（true，由sdk采集摄像头数据； false, 由客户端自己处理）
useSdkCollectAudio	boolean	非必填	默认值为false。（true，由sdk采集麦克风数据, false, 由客户端自己处理）
logSource	String	非必填	默认值为null。（日志来源，通常传appkey，同时用于控制是否上报sdk日志，默认不上报）
foregroundTimeOut	int	非必填	默认值为0。（前台超时时间，单位s）
backgroundTimeOut	int	非必填	默认值为0。（后台超时时间，单位s）
autoControlQuality	boolean	非必填	默认值为false。（控制Sdk是否允许依据网络情况自动切换码率及帧率，true，开启此功能；false，关闭此动能），功能开启后需要同时设置videoLevels才会生效，详见自动调整画质等级）
videoLevels	VideoParam[]	非必填	默认值null，(画质等级数组，只有autoControlQuality设置为true才会生效，开启后sdk会依据网络情况及传入的画质等级参数自动适配一组参数请求推流，详见自动调整画质等级）
h265StreamMode	int	非必填	默认值为1。（H265的设置方式：1-自动、2-手动、3-白名单模式）**2.6.0以上的版本才支持**
forceEncodeType	int	非必填	默认值为-1。在h265StreamMode等于2的情况下有效。（手动设置的推流模式：2-H264、10-H265）**2.6.0以上的版本才支持**

- **返回结果：**

通过BgsSdkCallback类中的**void onInitSuccess()**回调初始化成功。

通过BgsSdkCallback类中的**void onInitFail(int code, String msg)**回调初始化失败信息。

code：初始化失败错误码，msg变量中给出相应的初始化异常信息。

- **失败错误码**

错误码	错误描述
330102	初始化时uuid为空
330103	初始化时sdkView为空
330106	初始化时width为空
330107	初始化时height为空
330108	初始化时bitrate为空
330109	初始化时fps为空

云手机播放

> 特别说明：播放操作要确保在initGame成功后再进行。

startPhone

特别说明：播放操作要确保在initGame成功后再进行。

- **函数原型：**

```
public void startPhone(String serverToken)
```

- **函数说明:**连接并播放云手机。
- **返回结果：**

通过BgsSdkCallback类中的**void onConnectSuccess()**回调设备连接成功。

通过BgsSdkCallback类中的**void onConnectFail(int code, String msg)**回调连接和播放失败信息。

code：连接和播放失败错误码，msg变量中给出相应的连接和播放异常信息。

- **失败错误码**

错误码	错误描述
330101	serverToken为空
330201	连接设备失败
330202	调用startPhone接口时，initGame接口还没有回调onInitSuccess
330204	连接设备失败，服务端不可用，http status != 200的情况
330205	连接设备失败，服务端可用，http status == 200的情况，申请失败

pause

- **函数原型：**

```
public void pause()
```

- **函数说明：**暂停连接云手机，只有在调用startPhone后才生效，与resume对应。

resume

- **函数原型：**

```
public void resume()
```

- **函数说明：**恢复连接云手机，只有在调用startPhone后才生效，与pause对应。

stopPhone

- **函数原型：**

```
public void stopPhone()
```

- **函数说明：**释放云手机。

画质

setStreamConfig

- **函数原型：**

```
public void setStreamConfig(int width, int height, int bitrate, int fps)
```

- **函数说明：**设置视频宽高，帧率，码率，**须在第一帧画面返回后设置才能生效**。
- **参数说明：**

参数	类型	说明
width	int	压码视屏宽
height	int	压码视频高
bitrate	int	视频码率，单位kBps
fps	int	视频帧率

数据透传到云手机

sendKeyEvent

- **函数原型：**

```
public void sendKeyEvent(int action, int keyCode)
```

- **函数说明：**透传键值事件到云手机。
- **参数说明：**

参数	类型	说明
action	int	事件行为，按下或抬起，对应 KeyEvent.ACTION_DOWN和 KeyEvent.ACTION_UP
keyCode	int	具体按键值，传linux键值，对应KeyEvent.getScanCode()

sendSensorData

- **函数原型：**

```
public int sendSensorData(int type, float[] data)
```

- **函数说明：**透传传感器数据到云手机。

- **参数说明：**

参数	类型	说明
type	int	传感器类型，详见传感器类型
data	float[]	传感器数据

- **返回结果：**0成功，其他失败。

sendLocationData

- **函数原型：**

```
public int sendLocationData(float longitude,
                            float latitude,
                            float altitude,
                            float floor,
                            float horizontalAccuracy,
                            float verticalAccuracy,
                            float speed,
                            float direction,
                            String timestamp)
```

- **函数说明：**发送位置数据到云手机。

- **参数说明：**

参数	类型	说明
longitude	float	纬度(度)
latitude	float	经度(度)
altitude	float	高度(米)
floor	float	楼层
horizontalAccuracy	float	水平精度(米)
verticalAccuracy	float	垂直精度(米)
speed	float	速度(米/秒)
direction	float	方向
timestamp	String	时间戳

- **返回结果：**0成功，其他失败。

sendTransparentMsgReq

- **函数原型：**

```
public int sendTransparentMsgReq(int type, String data, String binderService)
```

- **函数说明：**透传信息到云手机。

- **参数说明：**

参数	类型	说明
type	int	自定义类型
data	String	具体数据
binderService	String	云端binder service名称

- **返回结果：**0成功，其他失败。

sendJoystickInput

- **函数原型：**

```
public int sendJoystickInput(int index, int pressed, int buttons,
                             int lx, int ly, int rx, int ry)
```

- **函数说明：**透传手柄键值到云手机。
- **参数说明：**

参数	类型	说明
index	int	手柄编号，从0开始递增
pressed	int	按键事件，抬起按下行为， 参见自定义手柄键值中ACTION_DOWN及ACTION_UP
buttons	int	按键值，参见自定义手柄键值
lx	int	左摇杆x坐标
ly	int	左摇杆y坐标
rx	int	右摇杆x坐标
ry	int	右摇杆y坐标

- **返回结果：**0成功，其他失败。

setExtraData

- **函数原型：**

```
public void setExtraData(int type, String info)
```

- **函数说明：**透传账号信息到云手机。
- **参数说明：**

参数	类型	说明
type	int	账号信息自定义类型
info	String	账号数据

sendAVData

- **函数原型：**

```
public int sendAVData(int avType, int frameType, byte[] data)
```

- **函数说明：**发送音视频数据到云手机，目前只支持H264及aac。
- **参数说明：**

参数	类型	说明
avType	int	用于区分音频（211）及视频（212），具体值参考传感器类型
frameType	int	音视频不同的编码类型，详情参考音视频编码帧类型
data	byte[]	具体音视频编码帧数据

- **返回结果：**0成功，其他失败。

copyToRemote

- **函数原型：**

```
public void copyToRemote(byte[] value)
```

- **函数说明：**透传剪切板数据到云手机的剪切板中，间接实现客户端到云手机的粘贴。
- **参数说明：**

参数	类型	说明
value	byte[]	剪切板的字符串数据要转为byte[]

sendInputString

- **函数原型：**

```
public void sendInputString(byte[] value)
```

- **函数说明：**透传字符串数据到云手机。
- **参数说明：**

参数	类型	说明
value	byte[]	字符串数据要转为byte[]

获取设备编号

getPadCode

- **函数原型：**

```
public String getPadCode()
```

- **函数说明：**获取设备编号，调用startGame后可获取。
- **返回结果：**返回设备编号。

音视频采集

setUseSdkCollectVideo

- **函数原型 ：**

```
public void setUseSdkCollectVideo(boolean isSdkCollectVideo)
```

- **函数说明：**控制是否由SDK采集摄像头数据并上传。
- **参数说明：**

参数	类型	说明
isSdkCollectVideo	boolean	true，由sdk采集摄像头数据； false, 由客户端自己处理

setAVEncodeParams

- **函数原型 ：**

```
public void setAVEncodeParams(AVEncodeParamsBean paramsBean)
```

- **函数说明：**设置sdk采集摄像头数据压码参数。
- **参数说明：**

参数	类型	说明
paramsBean	AVEncodeParamsBean	摄像头采集参数

openCamera

- **函数原型 ：**

```
public void openCamera()
```

- **函数说明：**打开摄像头并采集数据上传到云端，通常不需要主动调用此函数，只有当用户未授权摄像头权限，sdk回调通知用户去授权且等用户授权后才需要调用，权限请求接口回调详见BgsSdkCallback中**onRequestPermission**

setUseSdkCollectAudio

- **函数原型 ：**

```
public void setUseSdkCollectAudio(boolean isSdkCollectAudio)
```

- **函数说明：**控制是否由SDK采集麦克风数据并上传。
- **参数说明：**

参数	类型	说明
isSdkCollectAudio	boolean	true，由sdk采集麦克风数据, false, 由客户端自己处理

openMic

- **函数原型 ：**

```
public void openMic()
```

- **函数说明：**打开麦克风并采集数据上传到云端，通常不需要主动调用此函数，只有当用户未授权麦克风权限，sdk回调通知用户去授权且等用户授权后才需要调用，权限请求接口回调详见BgsSdkCallback中**onRequestPermission**

其他

getVersion

- **函数原型：**

```
public String getVersion()
```

- **函数说明：**获取sdk版本号。

- **返回结果**：返回sdk版本号。

audioPauseOrResume

- **函数原型** ：

```
public void audioPauseOrResume(boolean isResume)
```

- **函数说明**：控制sdk是否静音及恢复播放声音。
- **参数说明**：

参数	类型	说明
isResume	boolean	true，恢复播放声音；false，设置静音

sendKeyboardType

- **函数原型** ：设置云手机的键盘类型

```
public void sendKeyboardType(int type)
```

参数	类型	说明
type	int	0 是硬键盘 1 是软键盘

setNavBarVisible

- **函数原型** ：设置云手机的导航键状态

```
public void setNavBarVisible(boolean show)
```

参数	类型	说明
show	boolean	true: 导航键显示 false：导航键隐藏

getProtocolType

- **函数原型** ：获取当前的推流协议。
- **返回结果** ：返回当前的推流协议。

```
public String getProtocolType()
```

OPEN API参考

API文档

更新历史

发布时间	发布内容
2026-08-13	新增： 实例管理： 实例更换出口IP
	修改： 实例指令： 配置网络：proxyUser、proxyPassword调整为非必填

2026-07-21	上传文件到实例：自定义路径支持包含[.]或空格 任务管理： 任务结果查询：新增creater返回字段 应用管理： 预置应用卸载：支持通过packageName和suffix指定应用 内置应用更名为预置应用
2026-06-30	调整文档目录结构 新增： 实例管理 分页获取模板机编号 机型管理 分页查询机型信息 修改机型启用状态
2026-05-27	修改： 基础能力相关接口： 应用更新：新增checkAppPackage参数，支持校验传入包名与已有应用包名的一致性
2026-05-20	修改： 基础能力相关接口： 分页查询实例信息：新增storageType、aospDeployType、resourcePackageId、resourcePackageVersion、configId、softwarePackageUpdateTime字段
2026-05-07	修改： 基础能力相关接口： 分页查询物理机信息：支持查询子商户物理机设备
2026-04-15	新增： 基础能力相关接口： 获取实例预置应用安装列表 修改： 云手机相关接口： 获取serverToken、获取授权serverToken：增加idleTime入参 基础能力相关接口： 实例批量安装duapk应用镜像：增加forceInstall入参
2026-02-10	新增： 基础能力相关接口： 制作duapk镜像-高级模式 仿真授权码换绑应用
2026-01-12	新增： 基础能力相关接口：实例备份删除
2026-01-08	修改： 服务回调相关接口： 实例状态变更通知：新增实例故障状态的回调 基础能力相关接口 实例信息查询：增加instanceSn、deviceDiskModel、deviceMemoryModel、 deviceCpuModel字段
2025-12-04	新增： 服务回调相关接口： 物理机状态变更通知
2025-11-20	新增： 服务回调相关接口 实例状态变更通知

	实例状态变更通知
2025-11-06	修改： 基础能力相关接口 任务结果查询：增加taskType、createTime、executeTime、deviceIp返回字段 实例指令相关接口 实例重置：新增resetType入参 新增： 基础能力相关接口 获取任务类型列表 分页查询任务信息：可以根据任务ID或任务类型分页查询任务信息 应用清理缓存：支持以物理机维度清理应用缓存
2025-10-22	修改： 基础能力相关接口 实例截图：支持图片格式选择 云手机相关接口 获取serverToken：响应正在销毁的serverToken的error信息
2025-10-16	修改： 基础能力相关接口 实例备份：新增excludes、includes入参 新增： 基础能力相关接口 获取实例性能数据
2025-09-17	新增： 基础能力相关接口 实例部署营销套件 服务回调相关接口 物理机GPU告警通知 实例分配通知 实例一键新机通知 实例回收通知 资源告警通知 应用更新通知 实例部署营销套件通知
2025-09-03	新增： 基础能力相关接口 获取实例出口IP 获取实例应用安装列表 新增实例池 新增管理员账号 新增子商户
2025-08-20	新增： 基础能力相关接口 获取内置应用的桌面图标位置 配置内置应用的桌面图标位置 移除内置应用的桌面图标位置 修改： 基础能力相关接口 分页查询实例信息：新增物理机IP段 分页查询物理机信息：新增物理机IP段

2025-08-07	新增： 快照能力相关接口 快照配额设置 快照配额设置任务结果通知 修改： 基础能力相关接口 新增实例标签 编辑实例标签
2025-06-26	新增： 基础能力相关接口 分页查询实例标签 新增实例标签 编辑实例标签 删除实例标签 实例标签与实例关联 实例标签与实例关联解除 编辑实例自定义编号 分页查询告警策略 新增告警策略 更新告警策略 删除更新策略 启/禁用告警策略
2025-05-21	新增： 基础能力相关接口 实例对拷 服务回调相关接口 实例回调结果通知 消息栏消息通知
2025-05-06	新增： 基础能力相关接口 配置网络 快照能力相关接口 快照批量挂载
2025-04-17	新增： 基础能力相关接口 终止任务 分页查询物理机信息 带宽统计查询 实例指令相关接口 实例重启推流服务 快照能力相关接口 快照接口使用说明 快照初始化 快照卸载 快照挂载 分页查询快照记录 删除快照 快照初始化任务结果通知 快照卸载任务结果通知

	快照挂载任务结果通知
2025-04-02	新增： 基础能力相关接口 实例文件下载 服务回调相关接口 实例文件下载通知 实例会话受控状态变更通知
2025-03-19	新增： 云手机相关接口 实例批量解绑 基础能力相关接口 实例批量执行脚本
2025-03-06	新增： 基础能力相关接口 获取实例设备信息
2025-02-19	新增： 基础能力相关接口 实例故障率统计
2025-01-15	新增： 基础能力相关接口 物理机资源实时监控信息查询 物理机资源负载监控信息查询 实例资源实时监控信息查询 实例应用监控信息查询 修改实例维护状态 修改： 基础能力相关接口 分页查询实例信息 新增出参： maintainStatus 桌面推荐图标刷新：支持指定应用
2024-12-25	新增： 云手机相关接口 查询实例的所有绑定信息 解绑实例的所有绑定信息 基础能力相关接口 实例限速 分页查询应用列表 修改： 基础能力相关接口 分页查询实例信息 新增出参： instanceStatus deviceStatus
2024-10-30	新增： 基础能力相关接口 实例分辨率调整 服务回调相关接口 实例分辨率调整通知

	进程死亡告警通知
2024-10-16	新增： 基础能力相关接口 获取同步截图url 销毁已获取的同步截图url 系统镜像相关接口 删除自定义镜像
2024-09-25	新增： 基础能力相关接口 分辨率调整
2024-09-05	新增： 基础能力相关接口 应用安装 应用卸载 系统镜像相关接口 查询公共镜像 查询自定义镜像 上传自定义镜像 获取自定义镜像上传结果 创建更新镜像任务 查询更新镜像结果
2023-12-06	新增： 基础能力相关接口 实例切换会话控制权 实例设置内存限额
2023-11-02	新增： 基础能力相关接口 桌面推荐图标刷新 服务回调相关接口 桌面推荐图标刷新通知
2023-09-14	新增： 基础能力相关接口 实例授权连接 云手机相关接口 获取授权serverToken
2023-08-17	新增： 基础能力相关接口 应用上传(新版) 应用更新
2023-03-23	新增： 基础能力相关接口 实例信息查询[/resources/instance/infos] 下架： 基础能力相关接口 旧版分页查询实例信息接口[distribute/pad/infos.html]
2023-03-09	新增： 基础能力相关接口 客户实例汇总数据查询

	各个实例的总数据查询
2023-02-23	新增： 基础能力相关接口 创建流量查询任务接口 获取流量查询任务结果接口
2023-01-12	新增： 基础能力相关接口 机房已绑定实例数信息 实例执行脚本 上传文件到实例 服务回调相关接口： 实例执行脚本通知 上传文件到实例通知
2022-10-24	新增： 云手机相关接口 获取serverToken 新增入参： multiTerminal merchantPoolNo 实例指令相关接口 实例重启 新增入参： merchantPoolNo 实例重置 新增入参： merchantPoolNo
2022-10-20	新增： 基础能力相关接口 应用镜像按实例发布 应用镜像按机房发布 获取应用镜像按机房发布结果 服务回调相关接口 应用镜像机房发布完成通知
2022-08-18	新增接口： 基础能力相关接口 分页查询实例信息 云手机相关接口 获取severToken 实例解绑
2022-07-15	实例指令相关接口 实例解绑 服务回调相关接口 实例解绑回调
2022-06-10	新增接口 获取每个机房可用实例数量和总的可用实例数量

API概述

最近更新时间：2026年08月13日 12:00:00

云手机相关接口

接口名称	接口功能
获取serverToken	获取serverToken后可以用来申请实例。申请实例后可以用serverToken解绑已成功申请的实例。
获取授权serverToken	获取授权serverToken，通过serverToken连接的方式将云手机授权给其他用户使用
实例授权连接	将已连接的云手机授权给其他用户使用
新增实例到期时间	增加指定实例的到期时间。
更新实例到期时间	更新指定实例的到期时间。
查询实例的所有绑定信息	根据instanceCode查询实例的所有绑定信息。
实例解绑	销毁已申请的serverToken
实例批量解绑	批量销毁已申请的serverToken
解绑实例的所有绑定信息	销毁指定实例下已获取的所有serverToken

实例管理

接口名称	接口功能
分页查询实例信息	根据查询条件分页获取实例列表信息。
分页获取模板机编号	分页查询拥有的模板机，返回模板机的实例编号
获取实例设备信息	获取实例已安装的应用信息、可用磁盘总空间以及剩余磁盘空间
客户实例汇总数据查询	客户可以通过该接口来实时查询当前所有机房下的总实例数、可用实例数和已绑定的实例数。
修改实例类型	根据实例编码修改实例类型
修改实例维护状态	根据实例编号修改实例的维护状态
编辑实例自定义编号	编辑实例自定义编号
实例切换会话控制权	根据serverToken切换已绑定实例会话控制权。
机房未绑定实例数信息	获取每个机房可用且未绑定实例数量和总的可用且未绑定实例数量。
机房已绑定实例数信息	获取每个机房可用且已绑定实例数量和总的可用且已绑定实例数量。
获取ssh连接信息	获取ssh连接信息
获取实例出口IP	获取实例出口IP
实例更换出口IP	为指定实例发起更换出口IP任务
获取实例应用安装列表	获取实例应用安装列表
获取实例预置应用安装列表	获取实例预置应用安装列表

实例操作

接口名称	接口功能
实例重启	重启指定实例编码的实例
实例重置	将指定的实例恢复出厂设置
一键新机	一键新机
实例执行脚本	让实例执行传入的脚本内容
实例批量执行脚本	让多个实例执行传入的脚本内容
上传文件到实例	上传文件到百度云游戏平台
实例文件下载	提供实例内的文件下载地址
实例重启推流服务	实例重启推流服务
壁纸设置接口	针对不同的云手机实例，设置其壁纸。
实例分辨率调整	调整实例分辨率参数
实例设置内存限额	用于修改实例可以使用的内存上限值
实例限速	设置实例限速
桌面推荐图标刷新	用于刷新并展示实例桌面推荐应用，只能操作未绑定实例
配置网络	配置网络
实例对拷	实例对拷
实例截图	对实例进行截图，支持设置截图质量
实例截图结果查询	查询截图结果，如果截图成功，提供截图下载url
获取同步截图url	提供实例的即时截图的URL地址。
销毁已获取的同步截图url	销毁已获取的同步截图URL。
实例调度	对实例进行实例池之间的调度

实例池管理

接口名称	接口功能
新增实例池	新增实例池
实例池信息查询	查询实例池信息
实例池数量查询	查询实例池数量信息

标签管理

接口名称	接口功能
分页查询实例标签	分页查询实例标签
新增实例标签	新增实例标签
编辑实例标签	编辑实例标签
删除实例标签	删除实例标签
实例标签与实例关联	实例标签与实例关联
实例标签与实例关联解除	实例标签与实例关联解除

应用管理

接口名称	接口功能
分页查询应用列表	分页获取应用列表信息。
应用上传	上传应用到百度云游戏平台。
应用上传(新版)	上传应用到百度云游戏平台
应用更新	更新应用文件，应用信息不变更
应用删除	删除上传的应用
应用安装	为单台或多台云手机实例同时安装应用
应用卸载	为单台或多台云手机实例同时卸载应用
应用启停	根据实例编号和应用包名对实例进行应用启停的操作。执行的结果可以根据返回的任务ID,通过获取应用启停结果接口查看。
预置应用安装	为单台或多台云手机实例同时安装预置应用
预置应用卸载	为单台或多台云手机实例同时卸载预置应用
获取预置应用的桌面图标位置	获取预置应用的桌面图标位置
配置预置应用的桌面图标位置	配置预置应用的桌面图标位置
移除预置应用的桌面图标位置	移除预置应用的桌面图标位置
应用清理缓存	应用清理缓存

duapk

接口名称	接口功能
查询应用的duapk镜像	查询应用的duapk镜像
制作duapk镜像	制作duapk镜像
制作duapk镜像-高级模式	无需通过平台进行应用安装，直接基于实例实际安装的应用制作duapk镜像
移除应用的duapk镜像	移除应用的duapk镜像
上架duapk镜像	上架duapk镜像
下架duapk镜像	下架duapk镜像
分页查询上架的duapk信息	分页查询上架的duapk信息
[实例批量安装 duapk 应用镜像](#实例批量安装 duapk 应用镜像)	实例批量安装 duapk 应用镜像

系统监控

接口名称	接口功能
实例资源实时监控信息查询	根据code获取实例设备实时监控信息
实例应用监控信息查询	根据code获取查询时间范围内的实例设备应用CPU、MEM监控信息
获取实例性能数据	获取实例性能数据
实例故障率统计	获取时间范围内的实例故障率统计数据
物理机资源实时监控信息查询	根据物理机ip获取物理设备实时监控信息
物理机资源负载监控信息查询	根据物理机ip获取查询时间范围内的物理设备负载的监控信息
带宽统计查询	带宽统计查询
创建流量查询任务	创建流量查询任务
获取流量查询任务结果	获取流量查询任务的结果
分页查询告警策略	分页查询告警策略
新增告警策略	新增告警策略
更新告警策略	更新告警策略
删除告警策略	删除告警策略
启/禁用告警策略	启/禁用告警策略

任务管理

接口名称	接口功能
获取任务类型列表	获取任务类型列表
分页查询任务信息	分页查询任务信息
任务结果查询	根据任务id查询任务执行结果
终止任务	支持对状态为“待执行”和“准备中”的任务发起终止

系统镜像

接口名称	接口功能
查询公共镜像	查询系统中提供的公共镜像
查询自定义镜像	查询用户创建的自定义镜像
上传自定义镜像	允许用户上传自己的镜像到系统中
获取自定义镜像上传结果	查询自定义镜像上传的状态和结果
删除自定义镜像	删除已上传的自定义镜像
创建实例更新镜像任务	创建任务以更新实例使用的镜像
查询实例更新镜像结果	查询更新实例镜像任务的结果和状态
创建物理机更新镜像任务	创建任务以更新物理机上的系统镜像

物理机管理

接口名称	接口功能
分页查询物理机信息	分页查询物理机信息
物理机重启	重启指定物理机

商户管理

接口名称	接口功能
新增管理员账号	新增管理员账号
新增子商户	新增子商户

机型管理

接口名称	接口功能
分页查询机型信息	根据查询条件分页获取机型列表信息
修改机型启用状态	更新机型启用状态

备份还原

接口名称	接口功能
实例备份	对指定实例进行数据备份。
实例还原	将备份的数据还原到实例中。
实例备份删除	删除实例备份的快照数据。
分页查询实例备份信息	分页查询实例备份信息

快照能力

接口名称	接口功能
快照接口使用说明	说明快照接口的使用方法
快照初始化	初始化快照环境
快照卸载	卸载当前挂载的快照
快照挂载	挂载一个快照到指定实例
分页查询快照记录	分页查询系统中的快照记录
删除快照	删除指定的快照
快照初始化任务结果通知	通知用户快照初始化的结果
快照卸载任务结果通知	通知用户快照卸载的结果
快照挂载任务结果通知	通知用户快照挂载的结果
快照批量挂载	批量为实例挂载快照
快照配额设置	设置快照配额
快照配额设置任务结果通知	通知用户快照配额设置的结果

仿真授权管理

接口名称	接口功能
分页查询营销套件授权码	分页查询营销套件授权码
仿真授权码换绑应用	仿真授权码换绑应用
实例部署营销套件	实例部署营销套件

脚本自动化

接口名称	接口功能
新建自动化任务	新建自动化任务
查询自动化任务	查询自动化任务
取消自动化任务	取消自动化任务
查询自动化任务日志	查询自动化任务的执行日志

服务回调

接口名称	接口功能
回调接口使用示例	回调接口使用示例
应用安装通知	应用安装完成后的通知。
应用卸载通知	应用卸载完成后的通知。
保存应用通知	应用上传完成后的通知。
应用启动通知	应用启动完成后的通知。
应用停止通知	应用停止完成后的通知。
实例解绑通知	实例解绑完成后的通知。
实例重置通知	实例重置完成后的通知。
实例重启通知	实例重启完成后的通知。
实例执行脚本通知	实例完成执行脚本后的通知
上传文件到实例通知	文件上传到实例完成后的通知
桌面推荐图标刷新通知	桌面推荐图标刷新后的通知。
实例备份通知	实例备份完成后的通知。
实例还原通知	实例还原完成后的通知。
实例分辨率调整通知	分辨率调整完成后的通知
进程死亡告警通知	实例运行的应用进程非正常死亡后的告警通知
实例文件下载通知	实例文件下载任务执行完成后的结果通知
实例状态变更通知接口	实例状态变更通知接口
实例会话受控状态变更通知	实例会话受控状态变更通知
实例对拷结果通知	实例对拷结果通知
消息栏消息通知	消息栏消息通知
物理机GPU告警通知	物理机GPU告警通知
实例分配通知	实例分配通知
实例一键新机通知	实例一键新机通知
实例池数量变更通知	实例池数量变更通知
实例回收通知	实例回收通知
实例ROOT/UNROOT结果通知	实例ROOT/UNROOT结果通知
资源告警通知	资源告警通知
应用更新通知	应用更新通知
实例部署营销套件通知	实例部署营销套件通知
物理机状态变更通知	物理机状态变更通知

调用方式

请求结构

Headers参数说明

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

公共参数

公共参数用于标识请求用户并进行签名校验和用户鉴权，在每次请求时均需要携带公共参数才能正常发起请求。

注：后续接口文档中不再对公共参数进行说明

名称	类型	是否必须	备注	示例值
appkey	String	是	百度云游戏平台分配给第三方用户的接口 key	50b13132bb394901f151bc12
auth_ver	String	是	api 的版本号	3
nonce	Long	是	当前有的时间戳，精确到毫秒	1600846168490
s	String	是	方法签名，根据 appkey、auth_ver 和 nonce 计算	efabcaca7c66d629a639cee37d924813

方法签名生成规则

```
s = MD5(appkey + appkey值 + auth_ver + auth_ver值 + nonce + nonce值 + appSecret值)
```

注：字符串内容统一用 utf-8 字符集

方法签名生成示例

Java

```
String paramsStr = "appkey" + appKey +
      "auth_ver" + authVer +
      "nonce" + System.currentTimeMillis() +
      appSecret;//接口签名密钥（可配置）
String sign = org.apache.commons.codec.digest.DigestUtils.md5Hex(paramsStr.getBytes(StandardCharsets.UTF_8));
```

URL拼接公共参数示例

```
https://platform.armvm.com/{resourcePath}?s={s}&appkey={appkey}&auth_ver={auth_ver}&nonce={nonce}
```

body参数

参数名称	参数值	是否必须	示例	备注
msg	见msg加密规则	是		
createTime	公共参数中的nonce	是		

msg加密规则

加密模式：3DES——CBC模式，填充模式PKCS5Padding

加密步骤：

```
1.desKey截取前24位转byte[]
```

2.时间戳nonce转byte[]

3.接口请求实际入参的Json串转byte[]

4.3DES对内容进行加密（CBC模式，填充模式PKCS5Padding）

5.使用base64对加密结果进行编码

msg加密示例

Java

```
 // 填装实际接口请求参数
Map param = new HashMap();
param.put("param1", "value1");
param.put("param2", "value2");
param.put("param3", "value3");

// 请求参数转化为JSON字符串
String requestParam = new Gson().toJson(param);

// 请求参数加密
Cipher cipher = Cipher.getInstance("DESede/CBC/PKCS5Padding");
SecretKeyFactory skf = SecretKeyFactory.getInstance("DESede");
SecretKey secretKey = skf.generateSecret(new DESedeKeySpec(desKey.getBytes("utf-8")));
ByteBuffer byteBuffer = ByteBuffer.allocate(8);
byteBuffer.order(ByteOrder.BIG_ENDIAN);
byteBuffer.putLong(nonce);
byte[] iv = byteBuffer.array();
IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivParameterSpec);
byte[] cipherText = cipher.doFinal(requestParam.getBytes("utf-8"));
String msg = Base64.encodeBase64String(cipherText);
Map map = new HashMap();
map.put("createTime", nonce);
map.put("msg", msg);
// 转化为JSON字符串
String encryptBody = new Gson().toJson(map);
```

完整调用示例

Java

```
 package org.example.openapi;

import cn.hutool.json.JSONUtil;
import org.apache.commons.codec.digest.DigestUtils;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESedeKeySpec;
import javax.crypto.spec.IvParameterSpec;
import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.ByteBuffer;
import java.nio.ByteOrder;
import java.nio.charset.StandardCharsets;
import java.util.Base64;
import java.util.HashMap;
```

```
import java.util.HashMap;
import java.util.Map;

public class OpenApiTest {
    public static void main(String[] args) throws Exception {
        // 云游戏平台分配给第三方用户的接口Key
        String appKey = "4a8311be034943cfaa6a357fdc9f0461";
        // 接口版本号，现在有2和3两种版本的接口，具体按照接口文档上的版本号
        String authVer = "3";
        // 当前的时间戳
        long nonce = System.currentTimeMillis();
        // 接口签名秘钥
        String appSecret = "01a258aff68e6b2f2513df7f7dc3cf18";
        // 生成方法签名
        String paramsStr = "appkey" + appKey +
                "auth_ver" + authVer +
                "nonce" + nonce +
                appSecret;//接口签名密钥（可配置）
        String sign = DigestUtils.md5Hex(paramsStr.getBytes(StandardCharsets.UTF_8));
        // desKey 3DES秘钥（可配置），根据商户平台实际生成的替换此值
        String desKey = "cba76b7b0ab1011179087092effe1b8d";
        // 拼接公共参数到URL
        String apiUrl = "https://platform.armvm.com/{resourcePath}?s=" + sign + "&appkey=" + appKey + "&auth_ver=" +
authVer + "&nonce=" + nonce;

        // 填装实际接口请求参数
        Map<String, Object> param = new HashMap<>();
        param.put("param1", "value1");
        param.put("param2", "value2");
        param.put("param3", "value3");

        // 请求参数转化为JSON字符串
        String requestParam = JSONUtil.toJsonStr(param);

        // 请求参数加密
        Cipher cipher = Cipher.getInstance("DESede/CBC/PKCS5Padding");
        SecretKeyFactory skf = SecretKeyFactory.getInstance("DESede");
        SecretKey secretKey = skf.generateSecret(new DESedeKeySpec(desKey.getBytes(StandardCharsets.UTF_8)));
        ByteBuffer byteBuffer = ByteBuffer.allocate(8);
        byteBuffer.order(ByteOrder.BIG_ENDIAN);
        byteBuffer.putLong(nonce);
        byte[] iv = byteBuffer.array();
        IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
        cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivParameterSpec);
        byte[] cipherText = cipher.doFinal(requestParam.getBytes(StandardCharsets.UTF_8));
        String msg = Base64.getEncoder().encodeToString(cipherText);
        Map<String, Object> map = new HashMap<>();
        map.put("createTime", nonce);
        map.put("msg", msg);
        // 转化为JSON字符串
        String encryptBody = JSONUtil.toJsonStr(map);

        // 构造请求
        URL url = new URL(apiUrl);
        HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
        // 连接超时时间
        httpURLConnection.setConnectTimeout(1000);
        // 读取结果超时时间
        httpURLConnection.setReadTimeout(1000);
        httpURLConnection.setDoInput(true);
        httpURLConnection.setDoOutput(true);
        httpURLConnection.setUseCaches(false);
        httpURLConnection.setRequestProperty("Content-type", "application/json;charset=utf-8");
```

```
        httpURLConnection.setRequestMethod("POST");

        // 得到请求的输出流对象
        OutputStreamWriter out = new OutputStreamWriter(httpURLConnection.getOutputStream());
        // 把数据写入请求的Body
        out.write(encryptBody);
        out.flush();
        out.close();

        String result;
        try (InputStream in = 200 == httpURLConnection.getResponseCode() ? httpURLConnection.getInputStream() :
httpURLConnection.getErrorStream();) {
            byte[] buf = new byte[1024];
            int length;
            ByteArrayOutputStream bout = new ByteArrayOutputStream();
            while ((length = in.read(buf, 0, buf.length)) > 0) {
                bout.write(buf, 0, length);
            }
            bout.flush();
            result = new String(bout.toByteArray(), StandardCharsets.UTF_8);
        } finally {
            httpURLConnection.disconnect();
        }
        int responseCode = httpURLConnection.getResponseCode();
        if (responseCode == 200) {
            System.out.println("调用成功,接口返回结果:\n" + result);
        } else {
            System.out.println("调用失败");
        }
    }
}
```

Python

```
 import json
import time
import hashlib
import base64
import struct

import requests
from Crypto.Cipher import DES3
from Crypto.Util.Padding import pad


def _pad(data):
    padding_length = 8 - len(data) % 8
    padding = bytes([padding_length] * padding_length)
    return data + padding


def encrypt(key, iv, message):
    cipher = DES3.new(key[:24].encode('utf-8'), DES3.MODE_CBC, struct.pack('>Q', iv))
    padded_message = pad(message.encode('utf-8'), DES3.block_size)
    ciphertext = cipher.encrypt(padded_message)
    return ciphertext


def encrypt_with_base64(key, iv, message):
    ciphertext = encrypt(key, iv, message)
    return base64.b64encode(ciphertext).decode('utf-8')
```

```
if __name__ == '__main__':
    # 云游戏平台分配给第三方用户的接口Key
    app_key = "4a8311be034943cfaa6a357fdc9f0461"
    # 接口版本号，现在有2和3两种版本的接口，具体按照接口文档上的版本号
    auth_ver = "3"
    # 当前的时间戳
    nonce = int(time.time() * 1000)
    # 接口签名秘钥
    app_sec = "01a258aff68e6b2f2513df7f7dc3cf18"
    # 方法签名
    sign = "appkey%sauth_ver%snonce%d%s" % (app_key, auth_ver, nonce, app_sec)
    md5_sign = hashlib.md5(sign.encode('utf-8')).hexdigest()
    # desKey 3DES秘钥（可配置），根据商户平台实际生成的替换此值
    des_key = "cba76b7b0ab1011179087092effe1b8d"
    # 拼接公共参数到URL
    url = "https://platform.armvm.com/resourcePath?appkey=%s&auth_ver=%s&s=%s&nonce=%d" % (app_key, auth_ver,
md5_sign, nonce)
    data = {
        "param1": "value1",
        "param2": "value2",
        "param3": "value3",
    }
    msg = encrypt_with_base64(des_key, nonce, json.dumps(data))
    post_data = {
        "msg": msg,
        "createTime": nonce
    }
    print('url = ', url)
    print('data = ', post_data)
    res = requests.post(url, json.dumps(post_data), headers={
        'Content-Type': 'application/json'
    })
    print(res.text)
```

Go

```
 package main

import (
 "bytes"
 "crypto/md5"
 "encoding/base64"
 "encoding/hex"
 "encoding/json"
 "fmt"
 "github.com/forgoer/openssl"
 "net/http"
 "time"
)

func encryptWithBase64(key string, iv int64, message string) (string, error) {
 ivBytes := make([]byte, 8)
 for i := 0; i < 8; i++ {
  ivBytes[7-i] = byte(iv >> (8 * i))
 }
 ciphertext, err := openssl.Des3CBCEncrypt([]byte(message), []byte(key)[:24], ivBytes, openssl.PKCS7_PADDING)
 if err != nil {
  return "", err
 }
 return base64.StdEncoding.EncodeToString(ciphertext), nil
```

```
return base64.StdEncoding.EncodeToString(ciphertext), nil
}

func main() {
 appKey := "4a8311be034943cfaa6a357fdc9f0461"
 authVer := "3"
 nonce := time.Now().UnixNano() / int64(time.Millisecond)
 appSec := "01a258aff68e6b2f2513df7f7dc3cf18"

 sign := fmt.Sprintf("appkey%sauth_ver%snonce%d%s", appKey, authVer, nonce, appSec)
 md5Sum := md5.Sum([]byte(sign))
 md5Sign := hex.EncodeToString(md5Sum[:])

 desKey := "cba76b7b0ab1011179087092effe1b8d"

 url := fmt.Sprintf("https://platform.armvm.com/resourcePath?appkey=%s&auth_ver=%s&s=%s&nonce=%d", appKey,
authVer, md5Sign, nonce)

 data := map[string]string{
  "param1": "value1",
  "param2": "value2",
  "param3": "value3",
 }

 jsonData, err := json.Marshal(data)
 if err != nil {
  fmt.Println("Error marshalling JSON:", err)
  return
 }

 msg, err := encryptWithBase64(desKey, nonce, string(jsonData))
 if err != nil {
  fmt.Println("Encryption error:", err)
  return
 }

 postData := map[string]interface{}{
  "msg":        msg,
  "createTime": nonce,
 }

 postDataJson, err := json.Marshal(postData)
 if err != nil {
  fmt.Println("Error marshalling post data:", err)
  return
 }

 fmt.Println("url = ", url)
 fmt.Println("data = ", string(postDataJson))

 req, err := http.NewRequest("POST", url, bytes.NewBuffer(postDataJson))
 if err != nil {
  fmt.Println("Error creating request:", err)
  return
 }
 req.Header.Set("Content-Type", "application/json")

 client := &http.Client{}
 resp, err := client.Do(req)
 if err != nil {
  fmt.Println("Error making request:", err)
  return
 }
```

```
 defer resp.Body.Close()

 var responseBytes bytes.Buffer
 _, err = responseBytes.ReadFrom(resp.Body)
 if err != nil {
  fmt.Println("Error reading response:", err)
  return
 }

 fmt.Println(responseBytes.String())
}
```

云手机相关接口

获取serverToken

基本信息

URL

> https://platform.armvm.com/auth/instance/cloud-phone-server-token

请求格式

> POST

版本号

> 3

使用备注

> 同一个uuid+instanceCode的组合，只允许拥有一个有效的serverToken，不允许同时为相同的uuid+instanceCode申请多个serverToken。
>
> 如果传入的uuid+serverToken组合，已存在有效的serverToken，那么本接口会：
>
> 1. 如果已存在serverToken的控制权与本次传入的控制权，则返回错误：实例与uuid已存在绑定关系，且控制权不一致
> 2. 如果已存在serverToken正在销毁中，则返回错误：实例与uuid已存在绑定关系，但serverToken正在被销毁（这种情况可稍后重新申请）
> 3. 将已存在serverToken的到期时间与本次传入的onlineTime作比较，如果当前时间+onlineTime超过已存在serverToken的到期时间，则会自动进行续时操作，后将该serverToken返回给调用方
> 4. 其余情况，将把已存在的serverToken返回给调用方

请求参数

名称	类型	是否必须	默认值	备注	示例值
uuid	String	必须	无	用户唯一标识。uuid是用来区分真实用户的唯一标识	89e87993500e424ab0de340da6f3599e
instanceCodes	String[]	必须	无	实例编码集合，长度不能超过50	["VM192168169028","VM192168169029"]
onlineTime	Integer	必须	无	控制时长，单位：秒；1~2147483647	300
grantControl	String	必须	无	控制模式： CONTROL，可控制； WATCH：只能观看	WATCH
idleTime	Integer	非必须	无	控制空闲时长，单位：秒； 该入参会覆盖商户级别配置	15000

返回参数

名称	类型	是否必须	备注	示例值
code	number	非必须	响应码	0
data	object	非必须	返回信息	
├──successList	object[]	非必须	获取成功的列表	[{"instanceCode": "VM192168169028", "serverToken": "50b13132bb394901f151bc45"}]
├────instanceCode	String	非必须	实例编码	"VM192168169028"
├────serverToken	String	非必须	服务端token	"50b13132bb394901f151bc45"
├──errorList	object[]	非必须	获取失败的列表	[{"instanceCode": "VM192168169028", "errorMsg": "实例不存在"}]
├────instanceCode	String	非必须	实例编码	"VM192168169028"
├────errorMsg	String	非必须	错误信息	"实例不存在"
msg	string	非必须	响应消息	OK
ts	number	非必须	响应时间戳	1658393762049

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "data": {
    "successList": [
      {
        "instanceCode": "VM192168169027",
        "serverToken": "50b13132bb394901f151bc45"
      }
    ],
    "errorList": [
      {
        "instanceCode": "VM192168169028",
        "errorMsg": "实例不存在"
      },
      {
       "instanceCode": "VM192168169029",
       "errorMsg": "实例与uuid已存在绑定关系，且控制权不一致"
      },
      {
        "instanceCode": "VM192168169030",
        "errorMsg": "实例与uuid已存在绑定关系，但serverToken正在被销毁"
      }
    ]
  },
  "ts": "1658455442690"
}
```


异常示例


```
{
 "code": 11101118,
 "msg": "商户类型错误,只允许云手机客户使用",
 "ts": "1658455442690"
}
```


获取授权serverToken

基本信息

URL

> https://platform.armvm.com/auth/instance/authorized-server-token

请求格式

> POST

版本号

> 3

使用场景

> 用于实例授权场景，将已连接的云手机授权给其他用户使用，支持主用户和授权用户可同时使用。
>
> 注意：
>
> 1. 使用该接口前，需先通过获取serverToken接口，创建主serverToken
> 2. 通过该接口获取的serverToken，将会与主serverToken进行绑定，当主serverToken被销毁时，本接口获取的授权serverToken也会自动被销毁
> 3. 授权serverToken同样遵循“同一个uuid+instanceCode的组合，只允许拥有一个有效的serverToken”的限制

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
uuid	String	必须		用户唯一标识	89e87993500e424ab0de340da6f3599e
instanceCode	String	必须		实例编号	VM192168169028
onlineTime	Integer	必须		授权使用时间，单位：秒	3600
grantControl	String	必须		控制模式： CONTROL，可控制； WATCH：只能观看	CONTROL
idleTime	Integer	非必须		控制空闲时长，单位：秒； 该入参会覆盖商户级别配置	15000

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	必须			
├─serverToken	String	必须		服务端token	50b13132bb394901f151bc45
├─instanceCode	String	必须		实例编号	VM192168169028
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
     "instanceCode": "VM192168169028",
     "serverToken": "c5039adcda344cebb748e94644fbc68f5"
  },
  "msg": "OK",
  "ts": "1694584934947"
}
```

异常示例

```
{
  "code": 11103270,
  "msg": "实例暂未被使用，不能对该实例申请授权",
  "ts": 1601192809490
}
```

```
{
  "code": 11103271,
  "msg": "当前实例授权连接个数已达上限",
  "ts": 1601192809490
}
```

实例授权连接

基本信息

URL

> https://platform.armvm.com/auth/instance/authorized-connect

请求格式

> POST

版本号

> 3

使用场景

> 用于实例授权场景，将已连接的云手机授权给其他用户使用，支持主用户和授权用户可同时使用。
>
> 注意：
>
> 1. 使用该接口前，需先通过获取serverToken接口，创建主serverToken
> 2. 通过该接口获取的serverToken，将会与主serverToken进行绑定，当主serverToken被销毁时，本接口获取的授权serverToken也会自动被销毁
> 3. 授权serverToken同样遵循“同一个uuid+instanceCode的组合，只允许拥有一个有效的serverToken”的限制

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
uuid	String	必须		用户唯一标识	89e87993500e424ab0de340da6f3599e
instanceCode	String	必须		实例编号	VM192168169028
onlineTime	Integer	必须		授权使用时间，单位：秒	3600
grantControl	String	必须		控制模式： CONTROL，可控制； WATCH：只能观看	CONTROL

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	必须			
├─serverToken	String	必须		服务端token	50b13132bb394901f151bc45
├─instanceCode	String	必须		实例编号	VM192168169028
├─streamModeList	String[]	必须		推流方式配置	["1", "2"]
├─sessionId	String	必须		会话id	3WGVL2R7NQXUMKAAERANZXT003RNQRPZ
├─connData	String	必须		连接信息(json格式)	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例


```
{
  "code": 0,
  "data": {
    "connData": {
      "controlList": [
        {
          "controlCode": "TEST-TCP-CONTROL-242-bk",
          "controlInfoList": [
            {
              "cmccLine": "",
              "controlIp": "test.armvm.com",
              "controlPort": 10038,
              "ctLine": "",
              "cuLine": "",
              "traceServer": ""
            }
          ]
        }
      ],
      "multiControlList": [
        {
          "multiControlCode": "multi-242",
          "multiControlInfoList": [
            {
              "multiControlIp": "192.168.169.158"
```

```
                "multiControlIp": "192.168.169.138",
                "multiControlPort": 8080,
                "traceServer": ""
            }
        ]
    }
],
"padCode": "VM192168169028",
"remoteList": [],
"screenList": [
    {
        "screenCode": "SCREEN-GZ-TEST001",
        "screenInfoList": [
            {
                "bitrate": "50",
                "fps": "20",
                "gop": "5",
                "resolution": "2048",
                "screenIp": "127.0.0.1",
                "screenPort": 80,
                "traceServer": ""
            }
        ]
    }
],
"sessionExpireTime": "1694588533428",
"sessionId": "I0YUSI3Y9EX0BWF1JZF7XG4GG68ZE6VR",
"sessionInitTime": "1694584934428",
"webControlList": [
    {
        "webControlCode": "TEST-TCP-CONTROL-242-bk",
        "webControlInfoList": [
            {
                "cmccLine": "",
                "controlIp": "paas101.armvm.com",
                "controlPort": 10040,
                "ctLine": "",
                "cuLine": "",
                "traceServer": ""
            }
        ]
    }
],
"webRtcControlList": [
    {
        "controlCode": "TEST-TCP-CONTROL-242-bk",
        "gateway": {
            "cmccLine": "",
            "ctLine": "",
            "cuLine": "",
            "gatewayIp": "paas101.armvm.com",
            "gatewayPort": 8989
        },
        "webRtcControlInfoList": [
            {
                "controlIp": "192.168.170.242",
                "controlPort": 10038,
                "traceServer": ""
            }
        ]
    }
],
"webRtcMode": "",
```

```
        "webScreenList": [
          {
            "webScreenCode": "SCREEN-GZ-TEST001",
            "webScreenInfoList": [
              {
                "bitrate": "50",
                "fps": "20",
                "gop": "5",
                "resolution": "2048",
                "screenIp": "127.0.0.1",
                "screenPort": 10036,
                "traceServer": ""
              }
            ]
          }
        ]
      },
      "instanceCode": "VM192168169028",
      "serverToken": "c5039adcda344cebb748e94644fbc68f5",
      "sessionId": "I0YUSI3Y9EX0BWF1JZF7XG4GG68ZE6VR",
      "streamModeList": [
        "1",
        "2"
      ]
    },
    "msg": "OK",
    "ts": "1694584934947"
}
```


异常示例


```
{
    "code": 11103270,
    "msg": "实例暂未被使用，不能对该实例申请授权",
    "ts": 1601192809490
}
```



```
{
    "code": 11103271,
    "msg": "当前实例授权连接个数已达上限",
    "ts": 1601192809490
}
```


新增实例到期时间

基本信息

URL

> https://platform.armvm.com/resources/instance/expire-time-increase

请求格式

> POST

版本号

> 3

使用场景

> 按分钟数新增实例的到期时间

请求参数

名称	类型	是否必须	默认值	备注	示例值
serverTokens	String[]	必须	无	绑定的serverToken值	['021c6bee611b4d27aeb3de240d7144055']
time	Integer	必须	无	新增的时长，单位：分钟，范围：[1, 5256000]	1000

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	非必须			
├── failList	String[]	必须		续时失败的serverToken值	
msg	String	必须		返回信息	success
ts	Long	必须		接口返回时时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "failList": []
 },
 "code": 0,
 "msg": "OK",
 "ts": "1702454509182"
}
```

异常示例

```
{
 "code": 11102027,
 "msg": "serverToken无效",
 "ts": "1702458796121"
}
```

更新实例到期时间

基本信息

URL

> https://platform.armvm.com/resources/instance/expire-time-update

请求格式

> POST

版本号

> 3

使用场景

> 更新实例的到期时间
>
> 提示：本接口支持反向续时（将serverToken的到期时间修改为比原来更早的时间），反向续时过程中会中断正在进行的推流连接。

请求参数

名称	类型	是否必须	默认值	备注	示例值
serverTokens	String[]	必须	无	绑定的serverToken值	['021c6bee611b4d27aeb3de240d7144055']
expireTime	Long	必须	无	新的到期时间（时间戳）	1704038400000

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	非必须			
├── failList	String[]	必须		续时失败的serverToken值	
msg	String	必须		返回信息	success
ts	Long	必须		接口返回时时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "failList": []
 },
 "code": 0,
 "msg": "OK",
 "ts": "1702454509182"
}
```


异常示例


```
{
 "code": 11102027,
 "msg": "serverToken无效",
 "ts": "1702458796121"
}
```


查询实例的所有绑定信息

基本信息

URL

> https://platform.armvm.com/resources/instance/bind-infos

请求格式

> POST

版本号

> 3

使用场景

> 查询实例的所有绑定信息，即查询指定实例下有效的serverToken列表。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编码集合,集合长度在200内（包含200）	["VM192168001001","VM192168001002"]

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
├─instanceCode	String	非必须		实例编号	VM011101100190	
├─serverToken	String	非必须		serverToken	49b1f2943e04468383b10e4acafaad2d5	
├─uuid	String	非必须		用户uid	72f4d5bc0036	
├─businessType	String	非必须		绑定业务类型：cloud_phone-云手机；cloud_app-云游戏；oem-OEM	cloud_phone	
├─bindTime	Long	非必须		绑定时间	1734578634443	
├─expireTime	Long	非必须		到期时间	1735587244443	
├─controlStatus	String	非必须		受控状态：online-非受控，control-受控	online	
├─grantControl	String	非必须		控制权：CONTROL-可控制，WATCH-仅观看	CONTROL	
├─instanceControlTime	Long	非必须		最近受控时间	1734578795543	
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例


```
{
 "data": [
  {
   "instanceCode": "VM011101100190",
   "serverToken": "49b1f2943e04468383b10e4acafaad2d5",
   "uuid": "72f4d5bc0036",
   "businessType": "cloud_phone",
   "bindTime": 1734578634443,
   "expireTime": 1735587244443,
   "controlStatus": "online",
   "grantControl": "CONTROL",
   "instanceControlTime": null
  }
 ],
 "code": 0,
 "msg": "OK",
 "ts": "1734578646712"
}
```


异常示例

```
{
 "code": 11103062,
 "msg": "实例不存在",
 "ts": "1734578806958"
}
```


实例解绑

基本信息

URL

> https://platform.armvm.com/auth/instance/disconnect

请求格式

> POST

版本号

> 3

使用场景

> 销毁已申请的serverToken，销毁成功后将断开正在推流中的实例。
>
> 提示：销毁serverToken为异步过程，可监听实例解绑通知获取销毁结果。如果serverToken正在销毁中，无法再次使用相同的uuid+instanceCode申请serverToken，可在销毁成功后再次申请。

请求参数

名称	类型	是否必须	默认值	备注	示例值
uuid	String	必须	无	用户唯一标识。uuid是用来区分真实用户的唯一标识，必须与申请serverToken时的uuid一致	89e87993500e424ab0de340da6f3599e
serverToken	String	必须	无	申请实例时传入的serverToken	50b13132bb394901f151bc45

返回参数

名称	类型	是否必须	备注	示例值
code	number	非必须	响应码	0
data	object	非必须	返回信息	
msg	string	非必须	响应消息	OK
ts	number	非必须	响应时间戳	1658393762049

请求示例

> 参考：调用方式

响应示例


```
{
 "code": 0,
 "data": {},
 "msg": "OK",
 "ts": "1658455442690"
}
```


异常示例


```
{
 "code": 11102027,
 "msg": "serverToken无效",
 "ts": "1658455442690"
}
```


实例批量解绑

基本信息

URL

> https://platform.armvm.com/auth/instance/batch-disconnect

请求格式

> POST

版本号

> 3

使用场景

> 批量销毁已申请的serverToken，销毁成功后将断开正在推流中的实例。
>
> 提示：销毁serverToken为异步过程，可监听实例解绑通知获取销毁结果。如果serverToken正在销毁中，无法再次使用相同的uuid+instanceCode申请serverToken，可在销毁成功后再次申请。

请求参数

名称	类型	是否必须	默认值	备注	示例值
disconnectList	Object[]	必须	无	需要解绑的uuid和serverToken集合，最多200个	
├── uuid	String	必须	无	用户唯一标识。uuid是用来区分真实用户的唯一标识，必须与申请serverToken时的uuid一致	89e87993500e424ab0de340da6f3599e
├── serverToken	String	必须	无	申请实例时传入的serverToken	50b13132bb394901f151bc45

返回参数

名称	类型	是否必须	备注	示例值
code	number	非必须	响应码	0
data	object	非必须	返回信息	
├──successList	String[]	必须	成功的serverToken	["50b13132bb394901f151bc45"]
├──failList	Object[]	必须	失败的serverToken和原因	
├────────serverToken	String	必须	serverToken	50b13132bb394901f151bc46
├────────msg	String	必须	失败原因	serverToken无效
msg	string	非必须	响应消息	OK
ts	number	非必须	响应时间戳	1658393762049

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
   "successList": ["50b13132bb394901f151bc45"],
   "failList": [
     {
       "serverToken": "50b13132bb394901f151bc46",
       "msg": "serverToken无效"
     }
   ]
 },
 "msg": "OK",
 "ts": "1658455442690"
}
```

解绑实例的所有绑定信息

基本信息

URL

> https://platform.armvm.com/auth/instance/disconnect-all

请求格式

> POST

版本号

> 3

使用场景

> 销毁指定实例下已申请的所有serverToken，销毁成功后将断开正在推流中的实例。
>
> 提示：销毁serverToken为异步过程，可监听实例解绑通知获取销毁结果。如果serverToken正在销毁中，无法再次使用相同的uuid+instanceCode申请serverToken，可在销毁成功后再次申请。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编码集合,集合长度在200内（包含200）	["VM192168001001","VM192168001002"]

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object	非必须				
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {},
 "code": 0,
 "msg": "OK",
 "ts": "1734578060146"
}
```

异常示例

```
{
  "code": 11103062,
  "msg": "实例不存在",
  "ts": "1734578078843"
}

{
  "code": 11103397,
  "msg": "实例存在云游戏或OEM的绑定信息，暂不支持调用此接口解绑",
  "ts": "1734578136525"
}
```


实例管理

分页查询实例信息

基本信息

URL

> https://platform.armvm.com/resources/instance/infos

请求格式

> POST

版本号

> 3

使用场景

> 根据查询条件分页获取实例列表信息。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
page	Integer	非必须	1	页码，默认值1，最小1页（包含1页）	1
rows	Integer	非必须	200	每页记录数，默认值200，最小1条（包含1条），最大1000条（包含1000条）	10
instanceCodes	String[]	非必须		实例编号集合，集合长度在200内（包含200）	["VM19216800100 1"]
merchantPoolNos	Long[]	非必须		实例池编号集合，集合长度在50内（包含50）	[1, 2]
deviceIps	String[]	非必须		物理机IP集合，集合长度在200内（包含200）	["192.168.1.1"]
deviceIpSegment	String	非必须		物理机IP段	192.168.1
instanceType	String	非必须		实例类型：business-商业机；model-模板机；detect-检测机	business
instanceServerType	String	非必须		服务器类型	3588
romVersion	String	非必须		系统版本	android12.0
instanceGradeName	String	非必须		实例等级名称	VIP
idcCode	String	非必须		所属机房编号	GZNX-IDC-02
usableStatus	String	非必须		可用状态：usable-可用；unusable-不可用	usable
maintainStatus	Integer	非必须		维护状态：0-正常；1-维护	0
recycleStatus	String	非必须		释放状态：normal-正常；recycle-回收中；fail-失败	normal
taskStatus	String	非必须		任务状态：none-正常；executing-执行中	none
instanceStatus	Integer	非必须		实例在线状态：0:离线；1:在线	1
deviceStatus	Integer	非必须		物理机在线状态：0:离线；1:在线	1
malfunctionStatus	Integer	非必须		实例故障状态：0-正常；1-故障	0
networkStatus	Integer	非必须		网络状态：0-正常；1-欠费断网	0
bindStatus	String	非必须		绑定状态: bindable-未绑定；bound-已绑定	bindable
controlStatus	String	非必须		控制状态: online-在线；control-受控	online
imageVersionId	Long	非必须		系统镜像版本ID	1234
snapshotMountStatus	String	非必须		DuFS快照挂载状态：mounted-已挂载；unmounted-未挂载	mounted
snapshotId	Long	非必须		所挂载的DuFS快照ID	1234
tagId	Long	非必须		实例关联的标签ID	10001
emulationAuthStatus	Integer	非必须		仿真授权码绑定状态，0-未绑定，1-已绑定	0

返回参数

名称	类型	是否必须	备注	示例值
code	Integer	必须	响应编码	0
data	Object	非必须		

├──total	Integer	必须	总数	20
├──totalPage	Integer	必须	总页码	2
├──page	Integer	必须	页码	1
├──rows	Integer	必须	每页数量	10
├──pageData	Object[]	必须		
├────instanceCode	String	必须	实例编号	VM192168232001
├────instanceIp	String	必须	实例IP	192.168.232.1
├────deviceIp	String	必须	物理机IP	192.168.1.100
├────deviceIpSegment	String	必须	物理机IP段	192.168.1
├────deviceCode	String	必须	物理机编号	GZNX-IDC_DM011010099026
├────instanceType	String	必须	实例类型：business-商业机；model-模板机；detect-检测机	business
├────instanceServerType	String	必须	服务器类型	3588
├────instanceGrade	String	必须	开数	1
├────romVersion	String	必须	系统版本	android12.0
├────merchantPoolNo	Long	必须	商户实例池编号	1
├────idcCode	String	必须	机房编号	GZNX-IDC-02
├────idcName	String	必须	机房名称	南翔机房-可用区1
├────availableStatus	String	必须	可分配状态：unavailable-不可分配；available-可分配 备注：当可用状态为可用，且绑定状态为未绑定时，为可分配	available
├────usableStatus	String	必须	可用状态：usable-可用；unusable-不可用	usable
├────maintainStatus	Integer	必须	维护状态：0-正常；1-维护	0
├────recycleStatus	String	必须	释放状态：normal-正常；recycle-回收中；fail-失败	normal
├────taskStatus	String	必须	任务状态：none-正常；executing-执行中	none
├────instanceStatus	Integer	必须	实例在线状态：0:离线；1:在线	1
├────deviceStatus	Integer	必须	物理机在线状态：0:离线；1:在线	1
├────malfunctionStatus	Integer	必须	实例故障状态：0-正常；1-故障	0
├────networkStatus	Integer	必须	网络状态：0-正常；1-欠费断网	0
├────bindStatus	String	必须	绑定状态: bindable-未绑定；bound-已绑定	bindable
├────controlStatus	String	必须	控制状态: online-在线；control-受控	online
├────imageVersionId	Long	非必须	系统镜像版本ID	10001
├────imageVersionName	String	非必须	系统镜像版本名称	3588_aosp10_v2.25.1(2025-03-13)
├────snapshotMountStatus	String	必须	DuFS快照挂载状态：mounted-已挂载；unmounted-未挂载	mounted
├────snapshotId	Long	非必须	所挂载的DuFS快照ID	5001

├──────memoryLimit	Integer	非必须	内存限额，单位为MB，null则为不限额	1024
├──────upSpeed	Double	非必须	外网上行限速，单位为Mbps	100.0
├──────downSpeed	Double	非必须	外网下行限速，单位为Mbps	100.0
├──────intranetUpSpeed	Double	非必须	内网上行限速，单位为Mbps	10.0
├──────intranetDownSpeed	Double	非必须	内网下行限速，单位为Mbps	10.0
├──────allocateTime	Date	必须	调度时间（时间戳）	1733297481137
├──────recycleTime	Date	必须	回收时间（时间戳）	1733297481137
├──────tags	Object[]	非必须	实例关联的标签	
├────────────tagId	Long	必须	标签ID	10001
├────────────tagName	String	必须	标签名称	tag_name
├──────emulationAuthStatus	Integer	必须	仿真授权码绑定状态，0-未绑定，1-已绑定	0
├──────instanceSn	String	必须	同一物理机下实例的编号	0
├──────deviceCpuModel	String	必须	实例所属物理机CPU规格	8C
├──────deviceMemoryModel	String	必须	实例所属物理机内存规格	32GB
├──────deviceDiskModel	String	必须	实例所属物理机磁盘规格	512GB
├──────storageType	String	非必须	存储类型，取值：local/dufs/lvm/rbd/unknown	local
├──────aospDeployType	String	非必须	AOSP部署类型：0-unknown；1-普通共享；2-普通独立；3-overlayfs	1
├──────resourcePackageId	Long	非必须	资源包ID	10001
├──────resourcePackageVersion	String	非必须	资源包版本	v1.0.0
├──────configId	String	非必须	配置ID	config_123
├──────softwarePackageUpdateTime	Date	非必须	镜像更新完成时间（时间戳）	1733297481137
msg	String	必须	响应消息	
ts	Long	必须	时间戳	1733297481137

请求示例

> 参考：调用方式

响应示例


```
{
 "code": 0,
 "data": {
  "total": 20,
  "totalPage": 2,
  "page": 1,
  "rows": 10,
  "pageData": [
   {
    "instanceCode": "VM1021682220001",
```

```
        "instanceCode": "VM192168232001",
        "instanceIp": "192.168.232.1",
        "deviceIp": "192.168.1.100",
        "deviceIpSegment": "192.168.1",
        "deviceCode": "GZNX-IDC_DM011010099026",
        "instanceType": "business",
        "instanceServerType": "3588",
        "instanceGrade": "1",
        "romVersion": "android12.0",
        "merchantPoolNo": 1,
        "idcCode": "GZNX-IDC-02",
        "idcName": "南翔机房-可用区1",
        "availableStatus": "available",
        "usableStatus": "usable",
        "maintainStatus": 0,
        "recycleStatus": "normal",
        "taskStatus": "none",
        "instanceStatus": 1,
        "deviceStatus": 1,
        "malfunctionStatus": 0,
        "networkStatus": 0,
        "bindStatus": "bindable",
        "controlStatus": "online",
        "imageVersionId": 10001,
        "imageVersionName": "3588_aosp10_v2.25.1(2025-03-13)",
        "snapshotMountStatus": "mounted",
        "snapshotId": 5001,
        "memoryLimit": 1024,
        "upSpeed": 100.0,
        "downSpeed": 100.0,
        "intranetUpSpeed": 10.0,
        "intranetDownSpeed": 10.0,
        "allocateTime": 1733297481137,
        "recycleTime": 1733297481137,
        "tags": [
          {
            "tagId": "10001",
            "tagName": "tag_name"
          }
        ],
        "emulationAuthStatus": 0,
        "instanceSn": "0",
        "deviceCpuModel": "8C",
        "deviceMemoryModel": "32GB",
        "deviceDiskModel": "512GB",
        "storageType": "local",
        "aospDeployType": "1",
        "resourcePackageId": 10001,
        "resourcePackageVersion": "v1.0.0",
        "configId": "config_123",
        "softwarePackageUpdateTime": 1733297481137
      }
    ]
  },
  "msg": "success",
  "ts": 1733297481137
}
```


分页获取模板机编号

基本信息

URL

> https://platform.armvm.com/resources/instance/page-model-instance

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
page	Integer	是		分页查询页码	1
rows	Integer	是		分页查询每页行数，最大不能超过200	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	必须			
├── page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──total	Integer	必须		总数	20
├──totalPage	Integer	必须		总页码	2
├──pageData	Object []	必须			
├── ├──instanceCode	String	非必须		实例编号	VM192168001001
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "data": {
    "total": 20,
    "pageData": [
      {
        "instanceCode": "VM192168001001"
      },
      {
        "instanceCode": "VM192168001002"
      },
      {
        "instanceCode": "VM192168001003"
      }
    ],
    "rows": 10,
    "totalPage": 2,
    "page": 1
  },
  "ts": "1658455442690"
}
```


获取实例设备信息

基本信息

> 只支持查询到从平台发起安装的应用信息

URL

> https://platform.armvm.com/resources/instance/device-info

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	是		实例编号集合，最多只能传入200个	["VM192168001001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object[]	必须			
├──instanceCode	String	必须		实例编号	VM192168001001
├──storageSum	Double	必须		设备可用磁盘总空间，单位GB	100.0G
├──storageFree	Double	必须		设备剩余磁盘空间，单位GB	82.8G
├──appInstallList	Object[]	必须			
├── ├──appId	Long	必须		应用ID	1001
├── ├──appName	String	必须		应用名称	demo
├── ├──packageName	String	必须		应用包名	com.demo
├── ├──versionName	String	必须		应用版本名称	1.0.1
├── ├──appIconUrl	String	必须		应用图标地址	https://demo.com/xxx.jpg
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "data": [
    {
      "appInstallList": [
        {
          "appName": "demo",
          "packageName": "com.demo",
          "versionName": "1.0.1",
          "appId": 1001,
          "appIconUrl": "https://demo.com/xxx.jpg"
        }
      ],
      "storageSum": 100.0,
      "storageFree": 82.8,
      "padCode": "VM192168001001"
    }
  ],
  "ts": "1658455442690"
}
```

异常示例

```
{
  "code": 11100050,
  "msg": "系统异常，请联系客服人员"
}
```

客户实例汇总数据查询

基本信息

URL

> https://platform.armvm.com/resources/instance/summary-data

请求格式

> POST

版本号

> 3

使用场景

> 客户可以通过该接口来实时查询当前所有机房下的总实例数、可用实例数和已绑定的实例数。

请求参数

名称	类型	是否必须	默认值	备注	示例值
merchantPoolNo	Long	非必须	无	实例池编号	1
includeSubPool	Boolean	非必须	true	是否包含子实例池	true-包含子实例池,false-不包含子实例池

merchantPoolNo与includeSubPool的使用

merchantPoolNo	includeSubPool	筛选实例范围
null	null 或 true	当前商户的所有可用实例
null	false	当前商户顶级实例池中的可用实例
100	null 或 true	当前商户实例池编号为100的实例池及下层子实例池中的可用实例
100	false	当前商户实例池编号为100的实例池中的可用实例

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	必须			
├── totalData	Object	必须		实例总数信息	
├── ├── totalInstanceCount	Integer	必须		总数	10
├── ├── idcList	Object[]	必须		机房列表	
├── ├── ├── idcName	String	必须		机房名称	广州机房
├── ├── ├── idcCode	String	必须		机房编码	IDC-GZ
├── ├── ├── instanceCount	Integer	必须		数量	10
├── availableData	Object	必须		可用实例总数	
├── ├── totalInstanceCount	Integer	必须		总数	5
├── ├── idcList	Object[]	必须		机房列表	
├── ├── ├── idcName	String	必须		机房名称	广州机房
├── ├── ├── idcCode	String	必须		机房编码	IDC-GZ
├── ├── ├── instanceCount	Integer	必须		数量	5
├── boundData	Object	必须		已绑定实例总数	
├── ├── totalInstanceCount	Integer	必须		总数	
├── ├── idcList	Object[]	必须		机房列表	
├── ├── ├── idcName	String	必须		机房名称	广州机房
├── ├── ├── idcCode	String	必须		机房编码	IDC-GZ
├── ├── ├── instanceCount	Integer	必须		数量	5
msg	String	必须		返回信息	success
ts	Long	必须		接口返回时时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "success",
  "ts": 1658455442690,
  "data": {
    "totalData": {
      "totalInstanceCount": 10,
      "idcList": [
        {
          "idcName": "广州机房",
          "idcCode": "GZ-IDC-01",
          "instanceCount": 10
        }
      ]
    },
    "boundData": {
      "totalInstanceCount": 5,
      "idcList": [
        {
          "idcName": "广州机房",
          "idcCode": "GZ-IDC-01",
          "instanceCount": 5
        }
      ]
    },
    "availableData": {
      "totalInstanceCount": 5,
      "idcList": [
        {
          "idcName": "广州机房",
          "idcCode": "GZ-IDC-01",
          "instanceCount": 5
        }
      ]
    }
  }
}
```


修改实例类型

基本信息

URL

> https://platform.armvm.com/resources/instance/update-instance-type

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	备注	示例值
instanceCodes	String[]	是	最大200个	["VM192168043080"]
instanceType	String	是	实例类型	"business"

instanceType枚举

instanceType	含义
business	商业机
model	模板机
detect	检测机

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
msg	String	是		响应消息	"OK"
ts	Long	是		时间戳	1745494806123

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "OK",
 "ts": "1745494806123"
}
```

修改实例的维护状态

基本信息

URL

> https://platform.armvm.com/resources/instance/update-maintain-status

请求格式

> POST

版本号

> 3

使用场景

> 修改实例的维护状态

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号，最多200个	["VM192168001001","VM192168001002"]
maintainStatus	Integer	必须		维护状态，0-正常，1-维护	1

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应码	0
data	Object	非必须		响应码	
msg	String	必须		响应信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "OK",
 "ts": "1601188620078"
}
```

异常示例

```
{
 "code": 11103060,
 "msg": "实例不存在",
 "ts": "1601188620078"
}
```

编辑实例自定义编号

基本信息

URL

> https://platform.armvm.com/resources/instance/custom-code-update

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCustomList	Object[]	必须		单次请求最多1000个元素	
├─instanceCode	String	必须		实例编号	"VM192168001001"
├─customCode	String	必须		自定义编号，最长32个字符	"DB-01"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object[]	非必须			
├─instanceCode	String	必须			"VM192168001001"
├─errorMsg	String	必须			"实例不存在"
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1750839786752

请求示例

> 参考：调用方式

响应示例

```
{
 "data": [
  {
   "instanceCode": "VM192168001001",
   "errorMsg": "实例不存在"
  }
 ],
 "code": 0,
 "msg": "OK",
 "ts": "1750839786752"
}
```

异常示例

```
{
 "code": 11100011,
 "msg": "操作频繁，请稍后再试",
 "ts": "1750839786752"
}
```

实例切换会话控制权

基本信息

URL

> https://platform.armvm.com/resources/instance/session-control-switch

请求格式

> POST

版本号

> 3

使用场景

> 切换已绑定实例的会话控制权

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
watchServerTokens	String[]	非必须		需要切换成仅观看权限的serverToken集合；元素个数不能超过50	["39baa7d579034774af98da7cc449945c5","f209f26361bf487a9fdc88c88902753b5"]
controlServerTokens	String[]	非必须		需要切换成可控制权限的serverToken集合；元素个数不能超过50	["39baa7d579034774af98da7cc449945c5","f209f26361bf487a9fdc88c88902753b5"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应码	0
data	Object[]	非必须		切换失败的serverToken和对应的失败信息	
├─serverToken	String	必须		切换失败的serverToken	50b13132bb394901f151bc45
├─errorMsg	String	必须		切换失败原因	serverToken无效
msg	String	必须		响应信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "data": [
  {
   "serverToken": "39baa7d579034774af98da7cc449945c5",
   "errorMsg": "serverToken无效"
  },
  {
   "serverToken": "f209f26361bf487a9fdc88c88902753b5",
   "errorMsg": "同一个serverToken的权限切换期望不能同时为WATCH和CONTROL"
  }
 ],
 "code": 0,
 "msg": "OK",
 "ts": "1700721933096"
}
```


异常示例


```
{
  "code": 11100003,
  "msg": "watchServerTokens集合的元素个数不能大于50",
  "ts": "1700721933096"
}
```


机房未绑定实例数信息

基本信息

URL

> https://platform.armvm.com/monitor/pad/available-count.html

请求格式

> POST

版本号

> 3

请求参数

无

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──totalAvailableCount	Integer	非必须		可用实例总数	20
├──idcInfos	Object[]	非必须			
├────────idcName	String	必须		机房名称	广州机房
├────────idcCode	String	必须		机房编码	GZ-IDC-TEST-01
├────────availableCount	Integer	必须		可用数量	1001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例


```
{
"code": 0,
"data": {
  "totalAvailableCount": 128,
  "idcInfos": [
    {
      "idcName": "北京机房",
      "idcCode": "BJ-1001",
      "availableCount": 20
    },
    {
      "idcName": "广州机房",
      "idcCode": "GZ-10012",
      "availableCount": 108
    }
  ],
  "msg": "",
  "ts": 1601021167163
}
}
```


异常示例

机房已绑定实例数信息

基本信息

URL

> https://platform.armvm.com/monitor/pad/enable-bind-count.html

请求格式

> POST

版本号

> 3

请求参数

无

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──totalEnableBindCount	Integer	非必须		客户已绑定的设备数	20000
├──idcInfos	Object[]	非必须			
├────────idcName	String	必须		机房名称	广州机房
├────────idcCode	String	必须		机房编码	GZ-IDC-TEST-01
├────────enableBindCount	Integer	必须		机房的已绑定的设备数	999
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
   "totalEnableBindCount": 2000,
   "idcInfos": [
     {
       "idcName": "北京机房",
       "idcCode": "BJ-1001",
       "enableBindCount": 1001
     },
     {
       "idcName": "广州机房",
       "idcCode": "GZ-10012",
       "enableBindCount": 999
     }
   ],
   "msg": "",
   "ts": 1601021167163
 }
}
```

异常示例

获取ssh连接信息

基本信息

URL

> https://platform.armvm.com/resources/instance/ssh-info

请求格式

> POST

版本号

> 3

使用场景

> 获取ssh连接信息

请求参数

名称	类型	是否必须	默认值	备注	示例值
instanceCode	String	必须	无	实例编号	VM011101102170
connectType	Integer	必须	无	连接类型：1-ssh 2-adb	1
liveTime	Integer	必须	无	连接有效时间，单位：秒，范围：[1, 86400]	600

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	非必须			
├── sshInfo	Object	必须		ssh连接信息	
├── ├── instanceCode	String	必须		实例编号	
├── ├── sshCommand	String	必须		连接命令	
├── ├── sshPwd	String	必须		连接秘钥	
├── expireTime	Long	必须		过期时间戳	
msg	String	必须		返回信息	success
ts	Long	必须		接口返回时时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "sshInfo": {
   "instanceCode": "VM011101102170",
   "sshCommand": "ssh 11.101.102.170__62485__15920850610__1702456492@192.168.170.235 -p 17706",
   "sshPwd":
"XnoQiOuyAvTjL+YZG5pcM9qi4vOX1ZHga2EO4wBragMfD7NeDOJQPYjm4QRMJZkHDqFlVvufUYbpucGR0Tzvqg=="
  },
  "expireTime": "1702456492329"
 },
 "code": 0,
 "msg": "OK",
 "ts": "1702455888197"
}
```


异常示例


```
{
 "code": 11109015,
 "msg": "操作失败，实例不存在",
 "ts": "1702458767487"
}
```


获取实例出口IP

基本信息

URL

> https://platform.armvm.com/resources/instance/export-ip

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	是		实例编号集合，最多200个	["VM192168232001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─instanceCode	String	必须		实例编码	VM192168232001
├─publicIpAddress	String	必须		出口IP	192.1.1.1
├─networkOperator	String	非必须		出口IP对应运营商：电信、移动、联通、未知运营商	电信
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389326710

请求示例

> 参考：调用方式

响应示例

```
{
  "data": [
    {
      "instanceCode": "VM192168232001",
      "publicIpAddress": "192.1.1.1",
      "networkOperator": "电信"
    }
  ],
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```

异常示例

```
{
  "code": 11103062,
  "msg": "实例不存在",
  "ts": "1750389326710"
}
```

实例更换出口IP

基本信息

URL

> https://platform.armvm.com/resources/instance/change-public-ip

请求格式

> POST

版本号

> 3

使用说明

> 为指定实例发起更换出口IP的任务，分配商户专属IP池中的空闲线路。
>
> IP切换需要相应机房配置，如使用请先联系项目服务人员沟通开通事宜。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号集合，集合长度在200内（包含200）	["VM192168001001","VM1921680010(2"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object[]	非必须		任务结果列表	
├─instanceCode	String	必须		实例编号	VM192168001001
├─taskId	Long	必须		任务编号	2840204519134814256
msg	String	非必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": [
  {
   "instanceCode": "VM192168001001",
   "taskId": 2840204353481548485
  }
 ],
 "msg": "ok",
 "ts": 1663664302000
}
```

异常示例

```
{
 "code": 11103483,
 "msg": "IP切换失败，商户专属IP池无空闲线路",
 "ts": 1663664302000
}
```


获取实例应用安装列表

基本信息

URL

> https://platform.armvm.com/resources/instance/app-install-list

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	是		实例编号集合，最多200个	["VM192168232001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─instanceCode	String	必须		实例编码	VM192168232001
├─appInstallRecordList	Object[]	必须		安装应用列表	
├──appName	String	必须		应用名称	云手机
├──packageName	String	必须		应用包名	com.example.app
├──versionName	String	必须		版本名称	1.0
├──size	Long	必须		应用大小，单位Byte	102400
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389326710

请求示例

> 参考：调用方式

响应示例


```
{
  "data": [
    {
      "instanceCode": "VM192168232001",
      "appInstallRecordList": [
        {
          "appName": "云手机",
          "packageName": "com.example.app",
          "versionName": "1.0",
          "size": 102400
        }
      ]
    }
  ],
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```


异常示例


```
{
  "code": 11103062,
  "msg": "实例不存在",
  "ts": "1750389326710"
}
```


获取实例预置应用安装列表

基本信息

URL

> https://platform.armvm.com/resources/instance/preinstall-app-list

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	是		实例编号集合，最多200个	["VM192168232001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├──instanceCode	String	必须		实例编码	VM192168232001
├──appInstallRecordList	Object[]	必须		预置应用列表	
├────appName	String	必须		应用名称	云手机
├────packageName	String	必须		应用包名	com.example.app
├────versionName	String	必须		版本名称	1.0
├────size	Long	必须		应用大小，单位Byte	102400
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389326710

请求示例

> 参考：调用方式

响应示例

```
{
  "data": [
    {
      "instanceCode": "VM192168232001",
      "appInstallRecordList": [
        {
          "appName": "云手机",
          "packageName": "com.example.app",
          "versionName": "1.0",
          "size": 102400
        }
      ]
    }
  ],
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```

异常示例

```
{
  "code": 11103062,
  "msg": "实例不存在",
  "ts": "1750389326710"
}
```

实例操作

实例重启

基本信息

URL

> https://platform.armvm.com/command/pad/reboot.html

请求格式

> POST

版本号

> 3

使用场景

> 重启指定实例编码的实例

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
padCodes	String[]	非必须		实例编码集合,集合长度在200内（包含200）	["VM192168001001","VM192168001002"]
merchantPoolNos	Long[]	非必须		实例池编号集合,集合长度在50内（包含50）	[1,2]

说明

padCodes与merchantPoolNos不能同时为空;
padCodes为空，merchantPoolNos不为空时则重启merchantPoolNos的所有实例;

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
├─padCode	String	必须		实例编码	VM192168001001	
├─taskId	Long	必须		任务编号	2840204519134814256	
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
    "padCode": "VM192168041004",
    "taskId": "2840204353481548485",
 },
 "msg": "ok",
 "ts": 1663664302000
}
```


异常示例


```
{
 "code": -1,
 "msg": "实例编号不存在",
 "ts": 1663664302000
}
```


实例重置

基本信息

URL

> https://platform.armvm.com/command/pad/reset.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
padCodes	String[]	非必须		实例编码集合,集合长度在200内（包含200）	["VM192168001001","VM19216800 002"]
merchantPoolNos	Long[]	非必须		实例池编号集合,集合长度在50内（包含50）	[1,2]
resetType	String	非必须	DATA_ONLY	DATA_ONLY：默认只清理数据 IMAGE_ONLY：支持只恢复镜像 DATA_AND_IMAGE：支持同时恢复镜像、清理数据	DATA_ONLY

说明

padCodes与merchantPoolNos不能同时为空;

padCodes为空，merchantPoolNos不为空时则重启merchantPoolNos的所有实例;

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
├─padCode	String	必须		实例编码	VM192168001001	
├─taskId	Long	必须		任务编号	2840204519134814256	
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "padCode": "VM192168041004",
    "taskId": "2840204353481548485"
  },
  "msg": "ok",
  "ts": 1663664302000
}
```

异常示例

```
{
  "code": -1,
  "msg": "实例编号不存在",
  "ts": 1663664302000
}
```

一键新机

基本信息

URL

> https://platform.armvm.com/distribute/pad/new-pad.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
padModels	Object[]	必须		实例编码和机型集合，集合长度在200内（包含200）	
├─padCode	String	必须		实例编号	VM192168001001
├─imei	String	非必须			
├─serialno	String	非必须			
├─wifimac	String	非必须			
├─androidid	String	非必须			
├─model	String	非必须			
├─brand	String	非必须			
├─manufacturer	String	非必须			

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
├─padCode	String	必须		实例编码	VM192168001001	
├─taskId	Long	必须		任务编号	2840204519134814256	
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
    "padCode": "VM192168041004",
    "taskId": "2840204353481548485",
  },
 "msg": "ok",
 "ts": 1663664302000
}
```

异常示例

```
{
 "code": -1,
 "msg": "实例编号不存在",
 "ts": 1663664302000
}
```

实例截图

基本信息

URL

> https://platform.armvm.com/command/pad/execute-script.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
padCodes	String[]	必须		实例编号集合	["VM192168043080", "VM192168043091"]
scriptContent	String	必须		脚本内容	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1001
├─padCode	String	必须		实例编号	VM192168043091
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": [
  {
   "taskId": 1001,
   "padCode": "VM192168043080"
  },
  {
   "taskId": 1002,
   "padCode": "VM192168043091"
  }
 ],
 "msg": "",
 "ts": 1668572148662
}
```


异常示例


```
{
  "code":11103062,
  "msg":"实例不存在",
  "ts":"1668586771490"
}
```


实例批量执行脚本

基本信息

URL

> https://platform.armvm.com/resources/instance/batch-execute-script

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
scripts	Object[]	必须		实例编号和脚本内容集合，最多200个	
├── instanceCode	String	必须		实例编号	VM192168043080
├── scriptContent	String	必须		脚本内容	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	非必须			
├──successList	Object[]	必须		成功的实例编号和任务ID	
├────────instanceCode	String	必须		实例编号	VM192168043091
├────────taskId	Long	必须		任务ID	1001
├──failList	Object[]	必须		失败的实例编号和原因	
├────────instanceCode	String	必须		实例编号	VM192168043092
├────────msg	String	必须		失败原因	实例不存在
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "successList": [
      {
        "instanceCode": "VM192168043091",
        "taskId": 1001
      }
    ],
    "failList": [
      {
        "instanceCode": "VM192168043092",
        "msg": "实例不存在",
      }
    ]
  },
  "msg": "",
  "ts": 1668572148662
}
```

上传文件到实例

基本信息

URL

> https://platform.armvm.com/resources/instance/file-upload

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号集合	["VM192168043080", "VM192168043091"]
fileUrl	String	必须		文件URL	http://platform.armvm.com/test.apk
fileName	String	必须		文件名称	test.apk
fileMd5	String	必须		文件MD5摘要	185da05e944e869ebf077d6217d3efe9
autoInstall	Integer	非必须	0	枚举值：0,1，表示是否选择自动安装，**仅fileName参数值以.apk/.xapk结尾时生效**。 其中，0：不自动安装，1：自动安装 如果选择自动安装，在安装成功后，安装包会自动删除	0
customizeFilePath	String	非必须		上传到实例的自定义路径，该路径为/sdcard下的相对路径， 如该参数填/files，将上传至实例的/sdcard/files目录下	/tmp/files/

> 注：如果需要上传应用到实例并自动安装，需要fileName参数值以.apk/.xapk结尾，并且autoInstall参数值为1，安装成功后安装包会自动删除

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1001
├─instanceCode	String	必须		实例编号	VM192168043091
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "taskId": 1001,
      "instanceCode": "VM192168043080"
    },
    {
      "taskId": 1002,
      "instanceCode": "VM192168043091"
    }
  ],
  "msg": "",
  "ts": 1668572148662
}
```


异常示例


```
{
  "code": 11103062,
  "msg": "实例不存在",
  "ts": 1668586771490
}
```


实例文件下载通知

基本信息

URL

> https://platform.armvm.com/resources/instance/file-download

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceFiles	Object[]	必须		最多200个	
├── instanceCode	String	必须		实例编号；注意：在一次请求中实例编号不可重复，如果存在重复，则取最后一条	"VM192168001001"
├── filePath	String	必须		文件路径	"/sdcard/file.txt"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1000
├─instanceCode	String	必须		实例编号	"VM192168001001"
msg	String	必须		信息	"OK"
ts	Long	必须		时间戳	1742961939356

请求示例

> 参考：调用方式

响应示例

```
{
"code": 0,
"data": [
  {
    "taskId": 1000,
    "instanceCode": "VM192168001001"
  },
  {
    "taskId": 1001,
    "instanceCode": "VM192168001002"
  }
],
"msg": "OK",
"ts": 1742961939356
}
```

异常示例

```
{
  "code":11103062,
  "msg":"实例不存在",
  "ts":1742961939356
}
```

实例重启推流服务

基本信息

URL

> https://platform.armvm.com/resources/instance/reboot-remote-play

请求格式

> POST

版本号

> 3

使用场景

> 重启指定实例编码的实例推流服务

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编码集合,集合长度在200内（包含200）	["VM192168001001","VM192168001002"]

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
├─instanceCode	String	必须		实例编码	VM192168001001	
├─taskId	Long	必须		任务编号	2840204519134814256	
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
    "instanceCode": "VM192168041004",
    "taskId": "2840204353481548485",
  },
 "msg": "ok",
 "ts": 1663664302000
}
```

异常示例

```
{
 "code": -1,
 "msg": "实例编号不存在",
 "ts": 1663664302000
}
```

壁纸设置接口

基本信息

URL

> https://platform.armvm.com/resources/instance/set-wallpaper

请求格式

> POST

版本号

> 3

使用场景

> 针对不同的云手机实例，设置其壁纸。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String []	必须		云手机实例编号，最多支持100台设备	
wallpaperUrl	String	必须		壁纸图片的网络 URL 地址	
wallpaperMd5	String	必须		壁纸图片的 MD5 值（长度不超过 32）	
wallpaperSuffix	String	必须		壁纸图片的文件后缀（png, jpg 等常见图片格式，不用带点 '.'）	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	
data	Object []	非必须			
├── taskId	Long	必须		任务ID	
├── workflowId	String	必须		工作流 ID	
├── instanceCode	String	必须		实例编号	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
 "data": [
  {
   "taskId": "2615424541409882144",
   "workflowId": "2615424541409882137",
   "instanceCode": "VM011012193009"
  },
  {
   "taskId": "2615424541409882145",
   "workflowId": "2615424541409882137",
   "instanceCode": "VM011101100080"
  }
 ],
 "code": 0,
 "msg": "OK",
 "ts": "1761299505453"
}
```

异常示例

```
{
 "code": 11103072,
 "msg": "实例不存在",
 "ts": 1740572263582
}
```

实例分辨率调整

基本信息

URL

> https://platform.armvm.com/resources/instance/resolution

请求格式

> POST

版本号

> 3

注意

> 在使用1080p以上分辨率时，请确保您的设备的 manage-agent 版本为 4.27.0 或以上。您可以与 PMO 确认设备的 manage-agent 版本是否满足该要求。这样可以确保您的设备能够正常处理更高的分辨率。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String []	必须		实例编号列表，最多200个	["VM192168001001"]
height	Integer	必须		高	范围：300~4096
width	Integer	必须		宽	范围：300~4096
fps	Integer	必须		帧率	枚举：30/60/90
dpi	Integer	必须		dpi	范围：120~480

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object []	必须			
├─taskId	Long	必须		任务ID	100000
├─instanceCode	String	必须		实例编号	VM192168001001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM192168001001"
     }
  ],
  "msg": "",
  "ts": 1601188620078
}
```

异常示例

```
{
 "code": 11103062,
 "msg": "实例不存在",
 "ts": 1601021167163
}
```

实例设置内存限额

基本信息

URL

> https://platform.armvm.com/resources/instance/memory-limit

请求格式

> POST

版本号

> 3

使用场景

> 用于修改实例可以使用的内存上限值，修改内存限额会发生实例重启，请谨慎操作

请求参数

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号集合（最多200个）	["VM192168043080", "VM192168043091"]
memoryLimit	Integer	非必须		内存限额值，单位MB（isLimit为true时，必须设置限额值）	2048
isLimit	Boolean	必须		是否设置限额：true-限额 | false-不限额	true

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1001
├─instanceCode	String	必须		实例编号	VM192168043091
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "taskId": 1001,
      "instanceCode": "VM192168043080"
    },
    {
      "taskId": 1002,
      "instanceCode": "VM192168043091"
    }
  ],
  "msg": "",
  "ts": 1668572148662
}
```


异常示例


```
{
 "code":11103292,
 "msg":"实例内存限额不能为空",
 "ts":"1668584368093"
}
```


实例限速

基本信息

URL

> https://platform.armvm.com/resources/instance/set-speed

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编码集合,集合长度在200内（包含200）	["VM192168001001","VM192168001002"]
direction	String	必须		限速类型，up-上行限速，down-下行限速	up
speed	Double	必须		外网限速，范围为0.2-1024，也可传入0取消限速	0.20
intranetSpeed	Double	非必须		内网限速，范围为1-1024，也可传入0取消限速	1.00

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
├─instanceCode	String	必须		实例编码	VM192168001001	
├─taskId	Long	必须		任务编号	2840204519134814256	
msg	String	非必须		响应消息		
ts	Long	必须		时间戳	1598754678123	

请求示例

> 参考：调用方式

响应示例

```
{
"code": 0,
"data": {
    "instanceCode": "VM192168041004",
    "taskId": "2840204353481548485",
 },
"msg": "ok",
"ts": 1663664302000
}
```

异常示例

```
{
"code": -1,
"msg": "实例编号不存在",
"ts": 1663664302000
}
```

桌面推荐图标刷新

基本信息

URL

> https://platform.armvm.com/resources/instance/recommend-app-icon-refresh

请求格式

> POST

版本号

> 3

使用场景

> 用于刷新并展示实例桌面推荐应用。该接口会先重置桌面，因此限制只能操作未绑定实例
> 注：该功能只适用于OEM镜像的云手机

请求参数

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号集合	["VM192168043080", "VM192168043091"]
appIds	String[]	非必须		应用ID集合	["10000", "10001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1001
├─instanceCode	String	必须		实例编号	VM192168043091
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "taskId": 1001,
      "instanceCode": "VM192168043080"
    },
    {
      "taskId": 1002,
      "instanceCode": "VM192168043091"
    }
  ],
  "msg": "",
  "ts": 1668572148662
}
```


异常示例


```
{
 "code":11103287,
 "msg":"已绑定实例[VM192168043092]不能更新桌面推荐应用图标",
 "ts":"1668584368093"
}
```


配置网络

基本信息

URL

> https://platform.armvm.com/resources/instance/network-proxy-workflow-create

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	备注	示例值
networkProxyConfigs	Object[]	是	最大200个	
├── instanceCode	String	是	实例编码	VM011101001096
├── proxyHost	String	是	代理主机	127.0.0.1
├── proxyPort	Integer	是	代理端口	18080
├── proxyUser	String	否	代理用户名	user
├── proxyPassword	String	否	代理密码	password
├── proxyWhite	String	否	代理白名单,多个时用半角逗号分隔	user1,user2

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object[]	是			
├──instanceCode	String	是		实例编号	
├──taskId	Long	是		任务ID	
msg	String	是		响应消息	
ts	Long	是		时间戳	1745494806123

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM011101001096"
     }
  ],
  "msg": "",
  "ts": 1745494806123
}
```

异常示例

```
{
 "code": 11103353,
 "msg": "实例不存在或无操作权限",
 "ts": 1745496581163
}
```

实例对拷

基本信息

URL

> https://platform.armvm.com/resources/instance/data-copy

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
dataCopyList	Object[]	是		对拷实例列表，最多只能传入200个	
├── sourceInstanceCode	String	是		源实例编号	VM192168001001
├── targetInstanceCode	String	是		目标实例编号	VM192168001002
├── imei	String	否		最大长度32	
├── serialno	String	否		最大长度91	
├── wifimac	String	否		最大长度32	
├── androidid	String	否		最大长度91	
├── model	String	否		最大长度91	
├── brand	String	否		最大长度91	
├── manufacturer	String	否		最大长度91	
includes	String[]	否		要包含的目录(文件)，支持通配符*，最大长度50 **此目录需要对安卓目录结构，非常了解，如果客户未做这方面调研，建议设置为空字符串** 此目录为/data下的相对目录，不支持非/data目录下的include *符号表示此目录下所有层级子目录，如果只有一个，只会压缩一层目录 如果未填写includes，或者includes为空，则默认打包所有/data下的目录	["/aaa"]
excludes	String[]	否		要排除的目录(文件)，支持通配符*，最大长度50 同上，此目录为/data下的相对目录。可以使用通配符* 如果未填写excludes，或者excludes为空，则默认不排除任务目录	["/bbb"]
reset	Boolean	是		对拷成功后，源实例是否恢复出厂设置	true

注意事项：

1. 对拷的两个实例，要求所在机房、服务器类型、系统版本均一致
2. 对拷实例列表中，如果有传入机型信息，将在对拷完成后，目标实例设置为此机型信息，否则将设置为源实例的机型信息
3. 对拷实例列表中，如果有传入机型信息，那么机型的所有参数均不能为空（imei、serialno、wifimac、androidid、model、brand、manufacturer参数必须全部为空或者全部不为空）

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object[]	必须			
├─sourceInstanceCode	String	必须		源实例编号	VM192168001001
├─targetInstanceCode	String	必须		目标实例编号	VM192168001002
├─taskId	Long	必须		任务ID	1001
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "data": [
    {
      "sourceInstanceCode": "VM192168001001",
      "targetInstanceCode": "VM192168001002",
      "taskId": "1001"
    }
  ],
  "ts": "1658455442690"
}
```

异常示例

```
 {
   "code": 11103420,
   "msg": "实例VM192168001001和VM192168001002的服务器类型不一致",
   "ts": "1658455442690"
}
{
   "code": 11103421,
   "msg": "实例VM192168001001和VM192168001002的系统版本不一致",
   "ts": "1658455442690"
}
{
   "code": 11103422,
   "msg": "实例VM192168001001和VM192168001002的所在机房不一致",
   "ts": "1658455442690"
}
{
   "code": 11103424,
   "msg": "实例机型信息不完整",
   "ts": "1658455442690"
}
```


实例截图

基本信息

URL

> https://platform.armvm.com/command/pad/screenshot.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
padCodes	String[]	必须			["VM192168043080", "VM192168043091"]
quality	Integer	非必须	3	枚举值：1,2,3，表示截图质量。 其中，1：高质量，2：中质量，3低质量	3
pictureType	String	非必须	"png"	图片格式；枚举值：bmp、png	"bmp"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1001
├─padCode	String	必须		实例编号	VM192168043091
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
"code": 0,
"data": [
  {
    "taskId": 1001,
    "padCode": "VM192168043080"
  },
  {
    "taskId": 1002,
    "padCode": "VM192168043091"
  }
],
"msg": "",
"ts": 1668572148662
}
```

异常示例

```
{
  "code":11103062,
  "msg":"实例不存在",
  "ts":"1668586771490"
}
```

实例截图结果查询

基本信息

URL

> https://platform.armvm.com/command/pad/screenshot-info.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long []	必须		任务ID集合	[1001, 1002]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1001
├─padCode	String	必须		实例编号	VM192168043091
├─taskStatus	Integer	必须		截图任务状态，枚举值6，7，8。其中，6：截图中，7：截图成功，8：截图失败	7
├─url	String	必须		截图下载URL	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": [
  {
   "taskId": 1001,
   "padCode": "VM192168043080",
   "taskStatus": 7,
   "url":"https://platform.armvm.com/files/screenshot/6a8030673f534b9c948c58b51c615553.png"
  },
  {
   "taskId": 1002,
   "padCode": "VM192168043091",
   "taskStatus": 7,
   "url":"https://platform.armvm.com/files/screenshot/6a8030673f534b9c948c58b51c615555.png"
  }
 ],
 "msg": "",
 "ts": 1668572148662
}
```

异常示例

```
{
 "code":11109015,
 "msg":"操作失败，请输入正确的任务ID",
 "ts":"1668584368093"
}
```

获取同步截图url

基本信息

> 获取新的同步截图url，访问该url将从云手机实时获取截图

URL

> https://platform.armvm.com/resources/instance/get-screenshot-url

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号，最多200个	["VM192168169028"]
fullQuality	Integer	非必须	80	截图质量数值 范围[1-100]	100
scale	Integer	非必须	288	缩放比，等比缩放截图 0为不缩放 最小值：0 最大值：720	288
rotate	Integer	非必须	0	旋转角度 枚举：0，90，180，270	0
pictureType	String	非必须	"png"	图片格式；枚举值：bmp、png	"bmp"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	必须			
├──successList	Object[]	必须		成功的实例编号以及截图token	
├────────screenshotUrl	String	必须		截图地址	https://zc-sync-cap.armvm.com:9809/op-agent/pad/screenshot?time=1727581342254&route=%2BAiT1Lts4RASs6sBCRRYvg%3D%3D&screenshotToken=xxx
├────────instanceCode	String	必须		实例编号	VM192168169029
├──expireTime	Long	必须		token到期时间（时间戳）	1727667742254
├──failList	String[]	必须		失败的实例编号列表	["VM192168169028"]
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1727581342086

使用说明：

1. 每次获取的url有效期为10分钟，不可指定，可联系PMO调整每次获取url的有效期，最长不超过24小时
2. 使用GET请求screenshotUrl，将在云手机实时截取一张图片，并返回图片流
3. 截图状态码说明：

状态码	说明
200	成功
404、500	请求错误，包括：篡改过URL参数
400	无效的url
401	用户ID校验失败
402	url已过期
403	url已被销毁
501	执行截图程序失败，需要设备端排查
502	设备端程序端口不通
503	截图超时
504	设备ping不通
505	设备空间不足

4. 调用建议：将接口返回的screenshotUrl缓存起来，在到期或被销毁之前可一直请求，并且捕获402/403状态码以便重新获取新的同步截图url

请求示例

> 参考：调用方式

响应示例


```
{
  "data": {
    "successList": [
      {
        "instanceCode": "VM192168169029",
        "screenshotUrl": "https://zc-sync-cap.armvm.com:9809/op-agent/pad/screenshot?
time=1727581342254&route=%2BAiT1Lts4RASs6sBCRRYvg%3D%3D&screenshotToken=YEl6G8YsTUPcQW4cyauhXOt%2Fp

      }
    ],
    "failList": [],
    "expireTime": 1727667742254
  },
  "code": 0,
  "msg": "OK",
  "ts": "1727581342086"
}
```


异常示例


```
{
  "data": {
    "successList": [],
    "failList": [
      "VM192168169029"
    ],
    "expireTime": 1727582003554
  },
  "code": 0,
  "msg": "OK",
  "ts": "1727582003554"
}
```


销毁已获取的同步截图url

基本信息

> 销毁云手机当前所有的同步截图url

URL

> https://platform.armvm.com/resources/instance/destroy-screenshot-url

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCode	String	必须		实例编号	VM192168169028

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	必须			
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1727581342086

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {},
  "code": 0,
  "msg": "OK",
  "ts": "1727581342086"
}
```

异常示例

```
{
  "code": 11105017,
  "msg": "截图url不存在或已销毁",
  "ts": "1727586992599"
}
```

实例调度

基本信息

URL

> https://platform.armvm.com/resources/instance-pool/allocate

请求格式

> POST

版本号

> 3

使用场景

> 对实例进行实例池之间的调度

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
merchantPoolNo	Long	必须		目标实例池Id	10000
autoReset	Boolean	非必须	false	调度到子商户实例池时，是否自动执行恢复出厂设置	true
instanceCodes	String[]	必须		需要进行调度的实例编号；元素个数不能超过200	["VM192168001001","VM192168001002"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应码	0
msg	String	必须		响应信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "OK",
 "ts": "1601188620078"
}
```

异常示例

```
{
 "code": 11103180,
 "msg": "实例池不存在",
 "ts": "1601188620078"
}
```

新增实例池

基本信息

URL

> https://platform.armvm.com/resources/instance-pool/save

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
parentMerchantPoolNo	Long	非必须		父实例池编号，为空时则在当前商户的顶级实例池下新增，否则在指定的实例池下新增	10000
instancePoolName	String	必须		实例池名称，最长20个字符	test
instancePoolType	String	必须		实例池类型，cloud_app-云游戏，cloud_phone-云手机	cloud_phone

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─merchantPoolNo	Long	必须		新增的实例池编号	10001
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "merchantPoolNo": "10001"
 },
 "code": 0,
 "msg": "OK",
 "ts": "1750231776858"
}
```


异常示例


```
{
 "code": 11103180,
 "msg": "实例池不存在",
 "ts": "1750231743378"
}
```


实例池信息查询

基本信息

URL

> https://platform.armvm.com/resources/instance-pool/query

请求格式

> POST

版本号

> 3

使用场景

> 查询实例池信息

请求参数

body

名称	类型	是否必须	默认值	备注	示例值

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应码	0
data	Object[]	必须		实例池列表数据	
├── merchantPoolNo	Long	必须		商户实例池编号	10001
├── instancePoolName	String	必须		实例池名称	默认实例池
├── instancePoolType	String	必须		实例池类型（cloud_app-云游戏,cloud_phone-云手机）	
├── merchantId	Long	必须		商户 ID	20001
├── isSubMerchant	Boolean	必须		是否子商户	false
msg	String	必须		响应信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "OK",
 "ts": "1601188620078",
 "data": [
  {
   "merchantPoolNo": 10001,
   "instancePoolName": "",
   "instancePoolType": "",
   "merchantId": 20001,
   "isSubMerchant": false
  },
 ]
}
```

异常示例

```
{
 "code": 11103180,
 "msg": "实例池不存在",
 "ts": "1601188620078"
}
```

实例池信息查询

基本信息

URL

> https://platform.armvm.com/resources/instance-pool/count

请求格式

> POST

版本号

> 3

使用场景

> 查询实例池信息

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
merchantPoolNo	Long	必须		响应码	0

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应码	0
data	Integer	必须		实例池实例数量	1
msg	String	必须		响应信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "OK",
 "ts": "1601188620078",
 "data": 1
}
```

异常示例

```
{
 "code": 11103180,
 "msg": "实例池不存在",
 "ts": "1601188620078"
}
```

标签管理

分页查询实例标签

基本信息

URL

> https://platform.armvm.com/resources/instance-tag/page

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
tagName	String	非必须		标签名称	tag_name
tagId	Long	非必须		标签ID	10001
page	Integer	必须		页码	1
rows	Integer	必须		每页条数，最多1000	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──total	Integer	必须		总数	20
├──totalPage	Integer	必须		总页码	2
├──page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──pageData	Object []	必须			
├────────tagId	Long	必须		标签ID	10001
├────────tagName	String	必须		标签名称	tag_name
├────────relatedInstanceCount	Integer	必须		关联实例数	1
├────────belongToSubMerchant	Boolean	必须		是否归属子商户	false
├────────subMerchantName	String	非必须		归属子商户名称	
├────────remark	String	非必须		备注	
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例


```
{
 "data": {
  "page": 1,
  "rows": 10,
  "totalPage": 1,
  "total": 8,
  "pageData": [
   {
    "tagId": "10001",
    "tagName": "tagName_8de8d3881a79",
    "relatedInstanceCount": 1,
    "belongToSubMerchant": false,
    "subMerchantName": null,
    "remark": "remark_0ca421610057"
   }
  ]
 },
 "code": 0,
 "msg": "OK",
 "ts": "1750231517278"
}
```


新增实例标签

基本信息

URL

> https://platform.armvm.com/resources/instance-tag/save

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
tagName	String	必须		标签名称，最长100个字符，支持中英文、数字、点号、下划线或中划线，以大小写字母或者中文开头，不允许重复	tag_name
remark	String	非必须		备注，最长200个字符	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─tagId	Long	必须		标签ID	10001
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "tagId": "10001"
 },
 "code": 0,
 "msg": "OK",
 "ts": "1750231776858"
}
```

异常示例

```
{
  "code": 11103429,
  "msg": "已存在相同的标签名称",
  "ts": "1750231743378"
}
```


编辑实例标签

基本信息

URL

> https://platform.armvm.com/resources/instance-tag/update

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
tagId	Long	必须		标签ID	10001
tagName	String	必须		标签名称，最长100个字符，支持中英文、数字、点号、下划线或中划线，以大小写字母或者中文开头，不允许重复	tag_name
remark	String	非必须		备注，最长200个字符	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "data": null,
  "code": 0,
  "msg": "OK",
  "ts": "1750231776858"
}
```


异常示例


```
{
  "code": 11103430,
  "msg": "标签不存在",
  "ts": "1750232004298"
}
```


删除实例标签

基本信息

URL

> https://platform.armvm.com/resources/instance-tag/remove

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
tagIds	Long[]	必须		标签ID集合，最多200个	[10001]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例


```
{
"data": null,
"code": 0,
"msg": "OK",
"ts": "1750231776858"
}
```


异常示例


```
{
"code": 11103430,
"msg": "标签不存在",
"ts": "1750232004298"
}
```


实例标签与实例关联

基本信息

URL

> https://platform.armvm.com/resources/instance-tag/relate-add

请求格式

> POST

版本号

> 3

注意事项

> 单个实例支持关联最多50个标签。主商户实例请与主商户创建的标签关联、暂不支持主商户实例与子商户标签关联。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
tagIds	Long[]	必须		标签ID集合，最多50个	[10001]
instanceCodes	String[]	必须		实例编号集合，最多200个	["VM192168232001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": null,
 "code": 0,
 "msg": "OK",
 "ts": "1750231776858"
}
```

异常示例

```
{
 "code": 11103060,
 "msg": "实例不存在",
 "ts": "1750232754931"
}

{
 "code": 11103430,
 "msg": "标签不存在",
 "ts": "1750232004298"
}

{
 "code": 11103433,
 "msg": "待关联标签的归属子商户无选定实例的访问权限，无法进行关联",
 "ts": "1750232594300"
}
```

实例标签与实例关联解除

基本信息

URL

> https://platform.armvm.com/resources/instance-tag/relate-remove

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
tagIds	Long[]	必须		标签ID集合，最多50个	[10001]
instanceCodes	String[]	必须		实例编号集合，最多200个	["VM192168232001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "data": null,
 "code": 0,
 "msg": "OK",
 "ts": "1750231776858"
}
```

异常示例

```
{
 "code": 11103060,
 "msg": "实例不存在",
 "ts": "1750232754931"
}
```

应用管理

分页查询应用列表

基本信息

URL

> https://platform.armvm.com/resources/app/page

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Integer	非必须		应用ID	10001
appPackage	String	非必须		应用包名	com.example.app
page	Integer	必须		页码	1
rows	Integer	必须		每页条数	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─total	Integer	必须		总数	20
├─totalPage	Integer	必须		总页码	2
├─page	Integer	必须		页码	1
├─rows	Integer	必须		每页数量	10
├─pageData	Object[]	必须			
├──────appId	Integer	必须		应用id	10001
├──────appName	String	必须		应用名称	云手机
├──────appPackage	String	必须		应用包名	com.example.app
├──────versionName	String	必须		版本	1.0
├──────appTaskStatus	Integer	必须		应用任务状态；0-未同步，1-同步中，2-已同步，3-同步失败，4-文件下载中，5-文件下载失败，6-版本包名与应用包名不一致，7-安装包解析失败	1
├──────releaseMarketStatus	Integer	必须		是否已发布应用市场；0-未发布，1-已发布	1
├──────md5sum	String	必须		MD5值	697c836fb7df257b7e60f2436a504445
├──────totalInstallCount	Integer	必须		总安装数	100
msg	String	必须		响应消息	OK
ts	Long	必须		时间戳	1601021167163

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "total": 20,
    "totalPage": 2,
    "page": 1,
    "rows": 10,
    "pageData": [
      {
        "appId": 10001,
        "appName": "云手机",
        "appPackage": "com.example.app",
        "versionName":"1.0",
        "syncStatus":1,
        "releaseMarketStatus":1,
        "md5sum": "697c836fb7df257b7e60f2436a504445",
        "totalInstallCount": 100
      }
    ]
  },
  "msg": "",
  "ts": 1601021167163
}
```


异常示例


```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1601021167163
}
```


应用上传

基本信息

URL

> https://platform.armvm.com/distribute/apps/uploads.html

请求格式

> POST

版本号

> 3

接口说明

> 将指定的应用安装包文件上传至平台。 如果需要将应用直接安装至实例，请使用 上传文件到实例 接口。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
apps	Object[]	必须		应用数量，上限10个	
├── appId	Long	必须		应用ID	1
├── appName	String	必须		应用名称，字符长度最长为64	appTest
├── pkgName	String	必须		应用包名 ，字符长度最长为255	com.demo.app
├── url	String	必须		应用下载地址	https://www.demo.com/demo/download
├── md5sum	String	必须		应用文件的MD5	
taskId	Long	非必须		任务id，如果填写该参数，则在保存应用通知回调中携带该值	1

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	非必须		响应消息	ok
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
"code": 0,
"msg": "ok",
"ts": 1601021167163
}
```

异常示例

```
{
"code": 11103153,
"msg": "AppId不能重复",
"ts": 1601021202058
}
```

应用上传(新版)

基本信息

URL

> https://platform.armvm.com/resources/app/new-version

请求格式

> POST

版本号

> 3

接口说明

> 将指定的应用安装包文件上传至平台，如果应用ID已存在则更新新版本。 如果需要将应用直接安装至实例，请使用 上传文件到实例 接口。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Long	必须		应用ID	1
appName	String	必须		应用名称，字符长度最长为64	appTest
appPackageName	String	必须		应用包名 ，字符长度最长为255	com.demo.app
appUrl	String	必须		应用下载地址	https://www.demo.com/demo/download

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	非必须		响应消息	ok
data	Object	非必须		响应结果	
├── applicationVersionId	Long	必须		应用版本ID	
├── appId	Long	必须		应用ID	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "ok",
 "ts": 1601021167163
}
```


异常示例


```
{
 "code": 11103153,
 "msg": "AppId不能重复",
 "ts": 1601021202058
}
```


应用更新

基本信息

URL

> https://platform.armvm.com/resources/app/upgrade

请求格式

> POST

版本号

> 3

接口说明

> 对已上传至平台的应用进行更新。如需校验传入的包名与已有应用包名的一致性，可设置 checkAppPackage 为 true。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Long	必须		应用ID	1
appName	String	必须		应用名称，字符长度最长为64	appTest
appPackageName	String	必须		应用包名 ，字符长度最长为255	com.demo.app
appUrl	String	必须		应用下载地址	https://www.demo.com/demo/download
checkAppPackage	Boolean	非必须	false	是否校验传入的包名与已有应用的包名是否一致；true 时会校验 appPackageName 必须与系统中已存在的应用包名一致	false

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	非必须		响应消息	ok
data	Object	非必须		响应结果	
├── applicationVersionId	Long	必须		应用版本ID	
├── appId	Long	必须		应用ID	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "ok",
 "ts": 1601021167163
}
```

异常示例

```
{
 "code": 11103153,
 "msg": "AppId不能重复",
 "ts": 1601021202058
}

{
 "code": 11103486,
 "msg": "传入的应用包名与已存在的应用包名不一致",
 "ts": 1601021202058
}
```

应用删除

基本信息

URL

> https://platform.armvm.com/resources/app/delete

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appIds	Long[]	非必须		应用ID，最多200个	[10001]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	OK
ts	Long	必须		时间戳	1601021167163

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {},
  "msg": "",
  "ts": 1601021167163
}
```

异常示例

```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1601021167163
}

{
 "code": 11103386,
 "msg": "存在应用被云游戏试玩中，请移除后重试，应用ID：xxx",
 "ts": 1601021167163
}

{
 "code": 11103386,
 "msg": "应用存在可用的应用镜像，请先将这些应用的应用镜像删除后再重试，应用ID：xxx",
 "ts": 1601021167163
}
```

应用安装

基本信息

URL

> https://platform.armvm.com/resources/instance/install-app

请求格式

> POST

版本号

> 3

接口说明

> 将已上传到平台的应用安装到指定的实例上。 如果需要安装指定的安装包文件，请使用 上传文件到实例 接口。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Integer	必须		应用ID	10001
instanceCodes	String []	必须		云手机实例列表，最多200个	["VM192168001001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object []	必须			
├─taskId	Long	必须		任务ID	100000
├─instanceCode	String	必须		实例编号	VM192168001001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM192168001001"
     }
  ],
  "msg": "",
  "ts": 1601188620078
}
```


异常示例


```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1601021167163
}
```


应用卸载

基本信息

URL

> https://platform.armvm.com/resources/instance/uninstall-app

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Integer	非必须		应用ID（与包名二选一）	10001
packageName	String	非必须		应用包名（与 ID 二选一）	"com.foo"
instanceCodes	String []	必须		云手机实例列表，最多200个	["VM192168001001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├── taskId	Long	必须		任务ID	100000
├── instanceCode	String	必须		实例编号	VM192168001001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM192168001001"
     }
  ],
  "msg": "",
  "ts": 1601188620078
}
```

异常示例

```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1601021167163
}
```

应用启停

基本信息

URL

> https://platform.armvm.com/command/apps/app-operate.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
padCodes	String []	必须		实例编号，上限200个	VM192168169029
operateType	String	必须		枚举：start、stop、restart	start
packageName	String	必须		应用包名	com.demo.app

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object []	必须			
├─padCode	String	必须		实例编号	VM192168001001
├─taskId	Long	必须		任务id	10001
msg	String	非必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
    "total": 1,
    "pageData": [
       {
          "padCode": "VM192168232001",
          "padStatus": "0",
          "idcCode": "GZ-IDC-TEST-01",
          "padGrade": "1",
          "apps": [
             1000016516
          ]
       }
    ],
    "totalPage": 1,
    "page": 1,
    "rows": 10
 },
 "msg": "",
 "ts": 1601021167163
}
```

异常示例

```
{
 "code": -1,
 "msg": "实例[VM19216801100]不存在",
 "ts": 1601021202058
}
```

预置应用安装

基本信息

URL

> https://platform.armvm.com/resources/app/app-builtin-install

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Long	必须		应用ID	10001
instanceCodes	String []	必须		云手机实例列表，最多200个	["VM192168001001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object []	必须			
├─taskId	Long	必须		任务ID	100000
├─instanceCode	String	必须		实例编号	VM192168001001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM192168001001"
     }
  ],
  "msg": "",
  "ts": 1601188620078
}
```


异常示例


```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1601021167163
}
```


预置应用卸载

基本信息

URL

> https://platform.armvm.com/resources/app/app-builtin-uninstall

请求格式

> POST

版本号

> 3

接口说明

> 卸载实例内预置安装的应用，原理是删除实例中`/system/app`目录下的应用安装包文件，文件命名为`{应用包名}.{文件后缀}`。
> 接口入参 appId 和 packageName+suffix 二选一：
>
> - 传入appId时，平台会查询通过预置应用安装接口发起的预置安装记录，获取当前安装版本的应用包名和文件后缀
> - 传入packageName+suffix时，直接使用传入的应用包名和文件后缀
>   示例：传入packageName=`com.demo.app`，suffix=`apk`，则将实例中`/system/app/com.demo.app.apk`文件进行删除。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Long	非必须		应用ID，与packageName二选一	10001
packageName	String	非必须		应用包名，最长128字符，与appId二选一	com.demo.app
suffix	String	非必须		应用文件后缀，最长9字符；传入packageName时必填	apk
instanceCodes	String []	必须		云手机实例列表，最多200个	["VM192168001001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object []	必须			
├── taskId	Long	必须		任务ID	100000
├── workflowId	Long	必须		工作流ID	200000
├── instanceCode	String	必须		实例编号	VM192168001001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
        "taskId": 100000,
        "workflowId": 200000,
        "instanceCode": "VM192168001001"
    }
  ],
  "msg": "",
  "ts": 1601188620078
}
```

异常示例

```
{
  "code": 11103072,
  "msg": "应用不存在",
  "ts": 1601021167163
}
```

获取预置应用的桌面图标位置

基本信息

URL

> https://platform.armvm.com/resources/app/get-desktop-icon-config

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号，最多200个	["VM192168001001"]
appIds	Long[]	非必须		应用ID，最多200个	[1]
container	Integer	非必须		图标位置，-100正常区域，-101屏幕底栏	-100
screen	Integer	非必须		所在页码，从0开始	0
x	Integer	非必须		横坐标，范围0-3，以屏幕左上角为起点，向右递增	0
y	Integer	非必须		纵坐标，范围0-3，以屏幕左上角为起点，向下递增	0

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object[]	非必须			
├── instanceCode	String	必须		实例编号	VM192168001001
├── appId	Long	必须		应用ID	1
├── container	Integer	必须		图标位置，-100正常区域，-101屏幕底栏	-100
├── screen	Integer	必须		所在页码	0
├── x	Integer	必须		横坐标	0
├── y	Integer	必须		纵坐标	0
msg	String	必须		响应消息	OK
ts	Long	必须		时间戳	1601021167163

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": [
  {
   "instanceCode": "VM192168001001",
   "appId": 1,
   "container": -100,
   "screen": 0,
   "x": 0,
   "y": 0
  }
 ],
 "msg": "ok",
 "ts": 1663664302000
}
```


配置预置应用的桌面图标位置

基本信息

URL

> https://platform.armvm.com/resources/app/save-desktop-icon-config

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号，最多200个	["VM192168001001"]
appId	Long	必须		应用ID，应用必须先内置安装到实例中	1
container	Integer	必须		图标位置，-100正常区域，-101屏幕底栏	-100
screen	Integer	必须		所在页码，从0开始	0
x	Integer	必须		横坐标，范围0-3，以屏幕左上角为起点，向右递增	0
y	Integer	必须		纵坐标，范围0-3，以屏幕左上角为起点，向下递增	0
overwriteCoordinate	Boolean	非必须	false	当配置的坐标已存在其他的应用配置时，是否覆盖而不是抛异常	false

注意事项

1. 使用该功能需要云手机内置指定版本的桌面
2. 当container=-101时，y必须等于0，且screen必须等于x
3. 如果传入的实例中，该应用图标已经配置到桌面其它位置，调用该接口将会修改该应用的图标位置为传入的位置
4. 如果传入的实例中，该位置已被其它应用配置，将根据overwriteCoordinate值决定是覆盖还是抛异常，如果overwriteCoordinate=true，则覆盖，否则将抛异常
5. 配置图标位置之后，需清除云手机桌面数据才生效，可下发命令【pm clear 桌面包名】到云手机执行

桌面包名

安卓版本	桌面包名
6.0/8.1/15	com.android.launcher2
10	com.android.launcher3
12	com.android12.launcher3
13	com.android13.launcher3

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	OK
ts	Long	必须		时间戳	1601021167163

请求示例

参考：调用方式

响应示例

```
{
 "code": 0,
 "data": null,
 "msg": "ok",
 "ts": 1663664302000
}
```

异常示例

```
{
 "code": 11103439,
 "data": null,
 "msg": "实例未内置安装该应用",
 "ts": 1663664302000
}
```

移除预置应用的桌面图标位置

基本信息

URL

> https://platform.armvm.com/resources/app/remove-desktop-icon-config

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号，最多200个	["VM192168001001"]
appIds	Long[]	非必须		应用ID，最多200个，不传该参数则代表清除所有	[1]

注意事项

> 1. 使用该功能需要云手机内置指定版本的桌面
> 2. 配置图标位置之后，需清除云手机桌面数据才生效，可下发命令【pm clear 桌面包名】到云手机执行

桌面包名

安卓版本	桌面包名
6.0/8.1/15	com.android.launcher2
10	com.android.launcher3
12	com.android12.launcher3
13	com.android13.launcher3

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
msg	String	必须		响应消息	OK
ts	Long	必须		时间戳	1601021167163

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": null,
 "msg": "ok",
 "ts": 1663664302000
}
```

异常示例

```
{
 "code": 11103072,
 "data": null,
 "msg": "应用不存在",
 "ts": 1663664302000
}
```

应用清理缓存

基本信息

URL

> https://platform.armvm.com/resources/device/clean-app-cache

请求格式

> POST

版本号

> 3

接口说明

> 以物理机维度清理应用缓存，会同时清理物理机下所有实例的应用缓存

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
deviceIps	String[]	必须		物理IP列表，最多200个	["11.101.1.49"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├──taskId	Long	必须		任务ID	
├──deviceIp	String	必须		物理IP	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1762311419902

请求示例

> 参考：调用方式

响应示例

```
{
  "msg": "OK",
  "code": 0,
  "data": [
    {
      "taskId": "2813218757970419736",
      "deviceIp": "11.101.1.49"
    }
  ],
  "ts": 1762311419902
}
```

异常示例

```
{
 "code": 11103250,
 "msg": "物理机不存在，11.101.1.49",
 "ts": 1762311419902
}
```

duapk

查询应用的duapk镜像

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/page

请求格式

> POST

版本号

> 3

使用场景

> 查询应用的duapk镜像

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appImageId	String	非必须		镜像ID	12259080139067404
appImageStatus	String	非必须		镜像状态；creating-创建中,uploading-上传中,created-创建成功,checked-已验证,published-已发布,fail-创建失败,deleted-已删除	"created"
romVersion	String	非必须		系统版本	"android12.0"
appId	Long	非必须		应用ID	10001
appPackageName	String	非必须		应用包名	"com.example.app"
page	Integer	必须		页码	1
rows	Integer	必须		每页条数	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──total	Integer	必须		总数	1
├──totalPage	Integer	必须		总页码	1
├──page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──pageData	Object[]	非必须			
├────────appImageId	Long	必须		镜像ID	12259080139067404
├────────appUpdateVersion	String	必须		镜像版本	20250226201742
├────────appImageStatus	String	必须		镜像状态	"created"
├────────appImageName	String	非必须		镜像名称	
├────────appImageSize	Integer	非必须		镜像大小；单位：KB	10240
├────────romVersion	String	必须		系统版本	"android12.0"
├────────appId	Long	必须		应用ID	10001
├────────appPackageName	String	必须		应用包名	"com.example.app"
├────────createTime	Long	必须		创建时间	1740572263489
├────────idcSyncInfo	Object	非必须		机房同步信息	
├──────────idcName	String	必须		机房名称	"广州机房"
├──────────syncStatus	String	必须		同步状态	"syncing"
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1740572578363

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "page": 1,
  "rows": 10,
  "totalPage": 1,
  "total": 1,
  "pageData": [
   {
    "appImageId": 12259080139067404,
    "appUpdateVersion": 20250226201742,
    "appImageStatus": "created",
    "appImageName": null,
    "appImageSize": null,
    "romVersion": "android12.0",
    "appId": 51452191,
    "appPackageName": "com.example.app",
    "createTime": 1740572263489,
    "idcSyncInfo": [
     {
      "idcName": "广州机房",
      "syncStatus": "syncing"
     }
    ]
   }
  ]
 },
 "code": 0,
 "msg": "OK",
 "ts": 1740572578363
}
```


异常示例


```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1740572578363
}
```


制作duapk镜像

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/create

请求格式

> POST

版本号

> 3

使用场景

> 制作duapk镜像

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Long	必须		应用ID	10001
instanceCode	String	必须		实例编号	"VM000000000001"
autoUninstall	Integer	非必须		制作完成后是否自动卸载应用；0-不卸载，1-卸载	1

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─appImageId	Long	必须		镜像id	12259080139067404
├─createImageTaskId	Long	必须		制作镜像任务id	12259080139067402
├─uninstallAppTaskId	Long	非必须		卸载应用任务id	12259080139067403
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1740572263582

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
   "appImageId": 12259080139067404,
   "createImageTaskId": 12259080139067402,
   "uninstallAppTaskId": 12259080139067403
 },
 "code": 0,
 "msg": "OK",
 "ts": 1740572263582
}
```

异常示例

```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1740572263582
}
```

制作duapk镜像-高级模式

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/create-advanced

请求格式

> POST

版本号

> 3

使用场景

> 无需通过平台进行应用安装，直接基于实例实际安装的应用制作duapk镜像

使用说明

> 1、用于制作duapk镜像的实例类型必须为模板机；修改实例类型
> 2、单个物理机不支持同时制作多个duapk镜像；
> 3、单个商户同时制作duapk镜像的数量有一定限制；

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appPackageName	String	必须		应用包名	"com.example.cloudphone"
instanceCode	String	必须		模板机实例编号	"VM000000000001"
autoAppUninstall	Boolean	非必须	false	镜像制作完成后是否自动卸载应用	true
autoRelease	Boolean	非必须	false	镜像制作完成后是否自动上架	true
releaseMerchantPoolNos	Long[]	非必须	[]	上架的实例池范围编号	[1001, 1002]
displayName	String	非必须		应用上架展示名称	"我的应用"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─workflowId	Long	必须		工作流id	12259080139067404
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1770609600421

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "workflowId": 12259080139067404
 },
 "code": 0,
 "msg": "OK",
 "ts": 1770609600421
}
```

异常示例

```
{
 "code": 11103459,
 "msg": "当前物理机存在其它的应用镜像制作任务",
 "ts": 1770609600421
}
```

```
{
 "code": 11105004,
 "msg": "非模板机instanceCode=VM0000000000001",
 "ts": 1770609600421
}
```

```
{
 "code": 11108059,
 "msg": "当前任务类型发起数量超过上限[x]，请调整数量或稍后重试",
 "ts": 1770609600421
}
```

移除应用的duapk镜像

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/remove

请求格式

> POST

版本号

> 3

使用场景

> 移除应用的duapk镜像

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appImageIds	Long []	必须		镜像ID	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1740573137271

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "msg": "OK",
 "ts": 1740573137271
}
```

异常示例

```
{
 "code": 11103119,
 "msg": "应用镜像不存在",
 "ts": 1740573137271
}
```

上架duapk镜像

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/release

请求格式

> POST

版本号

> 3

使用场景

> 上架duapk镜像

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appImageId	Long	必须		镜像ID	12259080139067404
merchantPoolNos	Long[]	非必须		发布范围的实例池编号	
displayName	String	非必须		应用上架展示名称	"我的应用"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1740573508898

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {},
  "code": 0,
  "msg": "OK",
  "ts": 1740573508898
}
```

异常示例

```
{
  "code": 11103119,
  "msg": "应用镜像不存在",
  "ts": 1740573508898
}
```

下架duapk镜像

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/withdraw

请求格式

> POST

版本号

> 3

使用场景

> 下架duapk镜像

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appId	Long	非必须		应用ID	10001
appImageIds	Long[]	非必须		镜像ID	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1740574294348

请求示例

> 参考：调用方式

响应示例

```
{
"data": {},
"code": 0,
"msg": "OK",
"ts": 1740574294348
}
```

异常示例

```
{
"code": 11100085,
"msg": "操作的筛选条件不正确",
"ts": 1740574294348
}
```

分页查询上架的duapk信息

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/released-page

请求格式

> POST

版本号

> 3

使用场景

> 分页查询上架的duapk信息

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
appIds	Long[]	非必须		应用ID，上限1000个	[10001]
appName	String	非必须		应用名称	"云手机"
idcCode	String	非必须		机房编号	"GZ-TEST"
merchantPoolNo	Long	非必须		实例池编号	1000
appImageIds	Long[]	非必须		duapk镜像ID	1000
romVersion	String	非必须		系统版本	"android12.0"
page	Integer	必须		页码	1
rows	Integer	必须		每页条数，最大1000	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─total	Integer	必须		总数	20
├─totalPage	Integer	必须		总页码	2
├─page	Integer	必须		页码	1
├─rows	Integer	必须		每页数量	10
├─pageData	Object[]	非必须			
├──────duapkImagePublishId	Long	必须		上架ID	10000
├──────appId	Long	必须		应用ID	10001
├──────appName	String	必须		应用名称	"云手机"
├──────appPackageName	String	必须		应用包名	"com.example.app"
├──────appVersion	String	必须		应用版本	"1.0"
├──────appImageId	Long	必须		duapk镜像ID	12259080139067404
├──────appImageName	String	必须		duapk镜像名称	"com.example.cloudphone_1.9.6.06_VM000000000001_android12.0_20250626120000"
├──────appImageSize	Long	必须		duapk镜像大小	1024
├──────appUpdateVersion	Long	必须		镜像版本	20250226201742
├──────romVersion	String	必须		系统版本	"android12.0"
├──────merchantPoolNos	Long[]	非必须		发布范围的实例池编号	[1000]
├──────displayName	String	非必须		应用上架展示名称	"我的应用"
├──────excludePath	String[]	非必须		排除目录	
├──────dufsCdn	Object[]	非必须		cdn信息	
├──────idcCode	String	必须		机房编号	
├──────dufsCdnHost	String	必须		cdn host	
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1740573860675

请求示例

> 参考：调用方式

响应示例

```
{
 "data": {
  "page": 1,
  "rows": 10,
  "totalPage": 1,
  "total": 1,
  "pageData": [
   {
    "duapkImagePublishId": 10000,
    "appId": 10001,
    "appName": "云手机",
    "appPackageName": "com.example.app",
    "appVersion": "1.0",
    "appImageId": 12259080139067404,
    "appImageName": "com.example.cloudphone_1.9.6.06_VM000000000001_android12.0_20250626120000",
    "appImageSize": 1024,
    "appUpdateVersion": 20250226201742,
    "romVersion": "android12.0",
    "merchantPoolNos": [
     1000
    ],
    "excludePath": [],
    "dufsCdn": [
     {
      "idcCode": "GZ-TEST",
      "dufsCdnHost": "10.0.0.1:80"
     }
    ]
   }
  ]
 },
 "code": 0,
 "msg": "OK",
 "ts": 1740573860675
}
```


异常示例


```
{
 "code": 11103072,
 "msg": "应用不存在",
 "ts": 1740572263582
}
```


实例批量安装 duapk 应用镜像

基本信息

URL

> https://platform.armvm.com/resources/duapk-image/install-batch

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号（不超过 200）	['VM192168043080']
appImageId	String	必须		镜像 ID	
forceInstall	Boolean	非必须	false	强制安装	false

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─instanceCode	String	必须		实例编号	VM192168043091
├─taskId	Long	必须		任务 ID	1001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "taskId": "12361297649647620",
      "workflowId": "12361297649647617",
      "instanceCode": "VM011101100053"
    }
  ],
  "msg": "",
  "ts": 1668572148662
}
```

系统监控

实例资源实时监控信息查询

基本信息

URL

> https://platform.armvm.com/resources/instance-monitor-info

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例code列表，最多传入10个	["VM010107016082","VM010107016083"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object[]	非必须			
├──instanceCode	String	必须		实例编码	VM010107016082
├──instanceIp	String	必须		实例Ip	192.168.0.1
├──recordTime	Long	必须		记录时间戳	
├──cpuRate	Bigdecimal	必须		cpu使用率	12.40
├──gpuRate	Bigdecimal	必须		gpu使用率	
├──memRate	Bigdecimal	必须		内存使用率	
├──storageRate	Bigdecimal	必须		存储使用率	
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "data": [
    {
      "instanceIp": "192.168.0.1",
      "recordTime": "1736420400000",
      "cpuRate": 19.76,
      "gpuRate": 82.78,
      "memRate": 62.85,
      "storageRate": 9.52
    }
  ],
  "code": 0,
  "msg": "OK",
  "ts": "1736423215818"
}
```


异常示例


```
{
 "code": 11103403,
 "msg": "获取实例监控信息失败，请稍后再试",
 "ts": 1601021202058
}
```


实例应用监控信息查询

基本信息

URL

> https://platform.armvm.com/resources/instance-app-monitor-info

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCode	String	必须		实例code	"VM010107016082"
startTime	Date	必须		开始时间，整5分钟；时间格式：yyyy-MM-dd HH:mm:ss；最早查询时间间隔：90天	2025-01-15 12:00:00
endTime	Date	必须		结束时间，整5分钟；时间格式：yyyy-MM-dd HH:mm:ss；与开始时间最大相差30天	2025-01-15 12:05:00

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
date	Object[]	必须			
├──cpuRateTopTen	Object[]	必须		cpu使用率Top10	
├────appName	String	必须		app名称	app.test1
├────metricInfoList	Object[]	必须		监控信息	
├──────recordTime	Long	必须		记录时间戳	1736740800000
├──────value	BigDecimal	必须		使用率	10.08
├──memRateTopTen	Object[]	必须		内存使用率Top10	
├────appName	String	必须		app名称	
├────metricInfoList	Object[]	必须		监控信息	
├──────recordTime	Long	必须		记录时间戳	
├──────value	BigDecimal	必须		使用率	
Msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例


```
{
  "data": {
    "cpuRateTopTen": [
      {
        "appName": "app.test10",
        "metricInfoList": [
          {
            "recordTime": "1736740800000",
            "value": 73.15
          }
        ]
      },
      {
        "appName": "app.test11",
        "metricInfoList": [
          {
            "recordTime": "1736740800000",
            "value": 10.08
          }
        ]
      },
      {
        "appName": "app.test7",
        "metricInfoList": [
          {
            "recordTime": "1736740800000",
            "value": 7.77
          }
```

```
        ]
    },
    {
        "appName": "app.test8",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 8.24
            }
        ]
    },
    {
        "appName": "app.test9",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 21.48
            }
        ]
    },
    {
        "appName": "app.test3",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 9.36
            }
        ]
    },
    {
        "appName": "app.test4",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 14.7
            }
        ]
    },
    {
        "appName": "app.test5",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 12.45
            }
        ]
    },
    {
        "appName": "app.test6",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 7.78
            }
        ]
    },
    {
        "appName": "app.test1",
        "metricInfoList": [
            {
                "recordTime": "1736740800000",
                "value": 36.43
```

```
            }
          ]
        }
      ],
      "memRateTopTen": [
        {
          "appName": "app.test10",
          "metricInfoList": [
            {
              "recordTime": "1736740800000",
              "value": 73.15
            }
          ]
        },
        {
          "appName": "app.test11",
          "metricInfoList": [
            {
              "recordTime": "1736740800000",
              "value": 10.08
            }
          ]
        },
        {
          "appName": "app.test7",
          "metricInfoList": [
            {
              "recordTime": "1736740800000",
              "value": 7.77
            }
          ]
        },
        {
          "appName": "app.test9",
          "metricInfoList": [
            {
              "recordTime": "1736740800000",
              "value": 21.48
            }
          ]
        },
        {
          "appName": "app.test3",
          "metricInfoList": [
            {
              "recordTime": "1736740800000",
              "value": 9.36
            }
          ]
        },
        {
          "appName": "app.test4",
          "metricInfoList": [
            {
              "recordTime": "1736740800000",
              "value": 14.7
            }
          ]
        },
        {
          "appName": "app.test5",
          "metricInfoList": [
```

```
                {
                    "recordTime": "1736740800000",
                    "value": 12.45
                }
            ]
        },
        {
            "appName": "app.test6",
            "metricInfoList": [
                {
                    "recordTime": "1736740800000",
                    "value": 7.78
                }
            ]
        },
        {
            "appName": "app.test1",
            "metricInfoList": [
                {
                    "recordTime": "1736740800000",
                    "value": 36.43
                }
            ]
        },
        {
            "appName": "app.test2",
            "metricInfoList": [
                {
                    "recordTime": "1736740800000",
                    "value": 6.21
                }
            ]
        }
    ]
  },
  "code": 0,
  "msg": "OK",
  "ts": "1736738330433"
}
```


异常示例


```
{
 "code": 11103404,
 "msg": "获取实例应用监控信息失败，请稍后再试",
 "ts": 1601021202058
}
```


获取实例性能数据

基本信息

URL

> https://platform.armvm.com/resources/instance/metric-detail

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号,最大200个	["VM192168001001"]
recordTime	Date	必须		记录时间，时间格式：yyyy-MM-dd HH:mm；只能查询24小时内的数据	2025-10-16 12:00

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├──cpuUtilizationRate	BigDecimal			CPU利用率	
├──memoryUtilizationRate	BigDecimal			内存利用率	
├──storageUtilizationRate	BigDecimal			存储利用率	
├──ioReadSpeed	BigDecimal			IO读速度	
├──ioWriteSpeed	BigDecimal			IO写速度	
├──storageSum	BigDecimal			总存储空间大小(GB)	
├──storageFree	BigDecimal			剩余存储空间大小(GB)	
├──memorySum	BigDecimal			总内存空间大小(GB)	
├──memoryFree	BigDecimal			剩余内存空间大小(GB)	
├──cpuFreq	BigDecimal			CPU频率	
├──memoryFreq	BigDecimal			内存频率	
├──temperature	BigDecimal			设备温度	
├──cpuLoad	BigDecimal			CPU负载	
├──ioWait	BigDecimal			IO负载	
├──deviceUpTime	BigDecimal			设备运行时间	
├──gpuFreq	BigDecimal			GPU频率	
├──gpuUtilizationRate	BigDecimal			GPU利用率	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例


```
{
  "code": 0,
  "data": {
    "instanceCode": "VM192168001001",
    "cpuUtilizationRate": 45.75,
    "memoryUtilizationRate": 68.32,
    "storageUtilizationRate": 52.18,
    "ioReadSpeed": 125.5,
    "ioWriteSpeed": 89.25,
    "storageSum": 1024,
    "storageFree": 489.75,
    "memorySum": 64,
    "memoryFree": 20.25,
    "cpuFreq": 3.2,
    "memoryFreq": 2.4,
    "temperature": 42.5,
    "cpuLoad": 1.85,
    "ioWait": 0.35,
    "deviceUpTime": 3600.5,
    "gpuFreq": 1.55,
    "gpuUtilizationRate": 28.4
  },
  "msg": "",
  "ts": 1601188620078
}
```


异常示例


```
{
 "code": 11103062,
 "msg": "实例不存在",
 "ts": 1601021167163
}
```


实例故障率统计

基本信息

URL

> https://platform.armvm.com/monitor/instance/malfunction-statistics

请求格式

> POST

版本号

> 3

使用场景

> 获取时间范围内的实例故障率统计数据

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
startTime	String	必须		开始时间；yyyy-MM-dd HH:mm:ss	"2025-01-01 00:00:00"
endTime	String	必须		结束时间；yyyy-MM-dd HH:mm:ss	"2025-02-19 23:59:59"
timeUnit	String	必须		统计时间单位；分钟-MINUTE，小时-HOUR，天-DAY	"HOUR"
includeSubMerchant	boolean	非必须	false	是否包含子商户	true

timeUnit支持的时间跨度

timeUnit	时间跨度
MINUTE	一个自然日内
HOUR	两个自然月内
DAY	大于两个自然月

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应码	0
data	Object[]	非必须			
├─statisticsTime	string	必须		统计时间	"2025-01-22 18:00:00"
├─serverType	String	必须		设备类型	"3588"
├─instanceMalfunctionRate	Double	必须		故障率	0.2917
msg	String	必须		响应信息	
ts	Long	必须		时间戳	1738721318402

请求示例

> 参考：调用方式

响应示例

```
{
 "data": [
  {
   "statisticsTime": "2025-01-01 12:00:00",
   "serverType": "3588",
   "instanceMalfunctionRate": 0.2917
  },
  {
   "statisticsTime": "2025-01-01 12:00:00",
   "serverType": "845",
   "instanceMalfunctionRate": 1
  }
 ],
 "code": 0,
 "msg": "OK",
 "ts": "1738721318402"
}
```


异常示例


```
{
 "code": 1111113,
 "msg": "统计时间单位不正确",
 "ts": "1738721318402"
}
```


物理机资源实时监控信息查询

基本信息

URL

> https://platform.armvm.com/resources/device-monitor-info

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
deviceIps	String[]	是		物理机IP列表，最多传入10个	["192.168.0.1","192.168.0.2"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
date	Object[]	必须			
├──deviceIp	String	必须		物理机Ip	192.168.0.1
├──recordTime	Long	必须		记录时间戳	
├──cpuRate	Bigdecimal	必须		cpu使用率	12.40
├──gpuRate	Bigdecimal	必须		gpu使用率	
├──memRate	Bigdecimal	必须		内存使用率	
├──storageRate	Bigdecimal	必须		存储使用率	
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "data": [
    {
      "deviceIp": "11.20.103.62",
      "recordTime": "1736420400000",
      "cpuRate": 19.76,
      "gpuRate": 82.78,
      "memRate": 62.85,
      "storageRate": 9.52
    }
  ],
  "code": 0,
  "msg": "OK",
  "ts": "1736423215818"
}
```

异常示例

```
{
 "code": 11103402,
 "msg": "获取物理机监控信息失败，请稍后再试",
 "ts": 1601021202058
}
```

物理机资源负载监控信息查询

基本信息

URL

> https://platform.armvm.com/resources/device-monitor-info-query

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
deviceIps	String[]	必须		物理机Ip列表，最多传入10个	["192.168.0.1","192.168.0.2"]
startTime	Date	必须		开始时间，整小时；时间格式：yyyy-MM-dd HH:mm:ss；最早查询时间间隔：90天	2025-01-15 12:00:00
endTime	Date	必须		结束时间，整小时；时间格式：yyyy-MM-dd HH:mm:ss；与开始时间最大相差30天	2025-01-15 13:00:00

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
date	Object[]	必须			
├──deviceIp	String	必须		物理机Ip	
├──ioLoadRate	Object[]	必须		io负载率	
├────recordTime	Long	必须		记录时间戳	
├────value	BigDecimal	必须		负载率	
├──cpuLoadRate	Object[]	必须		cpu负载率	
├────recordTime	Long	必须		记录时间戳	
├────value	BigDecimal	必须		负载率	
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "data": [
    {
      "deviceIp": "11.20.103.62",
      "ioLoadRate": [
        {
          "recordTime": "1736236800000",
          "value": 23.81
        }
      ],
      "cpuLoadRate": [
        {
          "recordTime": "1736236800000",
          "value": 31.84
        }
      ]
    }
  ],
  "code": 0,
  "msg": "OK",
  "ts": "1736496642188"
}
```


异常示例


```
{
 "code": 11103402,
 "msg": "获取物理机监控信息失败，请稍后再试",
 "ts": 1601021202058
}
```


带宽统计查询

基本信息

URL

> https://platform.armvm.com/monitor/network-bandwidth/list

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
idcCode	String	非必须		机房编号	GZNX-IDC-02
statUnit	String	必须		统计单位：Mbps/bps	bps
beginTime	String	必须		开始时间：yyyy-MM-dd HH:mm:ss	2025-03-19 00:05:00
endTime	String	必须		结束时间：yyyy-MM-dd HH:mm:ss	2025-03-19 17:10:00

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
date	Object[]	必须			
├──peek	Double	必须		峰值	4335.83
├──peek95	Double	必须		95峰值	4034.76
├──average	Double	必须		平均带宽	3367.68
├──bandwidthList	Object[]	非必须		带宽明细列表，不指定机房编号查询时返回空	
├────────recordTime	String	必须		记录时间：yyyy-MM-dd HH:mm:ss	2025-03-19 17:10:00
├────────send	Double	必须		发送数据	3321.62
├────────receive	Double	必须		接收数据	3960.42
ts	Long	必须		接口返回时时间戳	1744287887367

请求示例

> 参考：调用方式

响应示例


```
{
 "data": {
  "peek": 4335.83,
  "peek95": 4034.76,
  "average": 3367.68,
  "bandwidthList": [
   {
    "recordTime": "2025-03-19 17:10:00",
    "send": 3321.62,
    "receive": 3960.42
   },
   {
    "recordTime": "2025-03-19 17:05:00",
    "send": 3362.98,
    "receive": 3396.23
   }
  ]
 },
 "code": 0,
 "msg": "OK",
 "ts": "1744287887367"
}
```


异常示例

```
{
 "code": 1111112,
 "msg": "获取带宽流量数据失败，请稍后再试",
 "ts": 1744287887367
}
```


创建流量查询任务

基本信息

URL

> https://platform.armvm.com/distribute/task/query-flow

请求格式

> POST

版本号

> 3

使用场景

> 创建流量查询任务，查询给定实例集合的流量或宽带计费汇总，查询时间范围限制为只能查询60分钟以前60天以内的数据。
>
> 具有最大任务创建数限制，超过限制时，需要等待前面的任务执行完毕后才能创建新的查询任务。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例列表集合	["VM011021103175"]
startTime	String	必须		查询开始时间（必须为小时整点时间），格式为yyyy-MM-dd hh:mm:ss	2023-02-23 18:00:00
endTime	String	必须		查询结束时间（必须为小时整点时间），格式为yyyy-MM-dd hh:mm:ss，	2023-02-23 18:00:00
billingType	String	必须		计费类型，枚举类型：FLOW（流量计费）、PEEK_PERCENT_100（峰值带宽计费）、PEEK_PERCENT_95（95峰值带宽计费）	FLOW
taskDesc	String	非必须		任务描述	"2023年2月份流量计费汇总"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	必须			
├── taskId	Long	必须		任务ID	89695
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "data": [
    {
      "taskId": "89695",
    },
  ],
  "ts": "1658455442690"
}
```

异常示例

```
{
  "code": 11100064,
  "msg": "查询开始时间不能早于前60天",
  "ts": 1601192809490
}
```

```
{
  "code": 11100065,
  "msg": "查询结束时间只能到60分钟以前",
  "ts": 1601192809490
}
```

```
{
  "code": 11109021,
  "msg": "当前已有%s个查询任务正在执行，超过任务个数限制，请稍后再试",
  "ts": 1601192809490
}
```

```
{
  "code": 11100050,
  "msg": "系统异常，请联系客服人员",
  "ts": 1601192809490
}
```

获取流量查询任务结果

基本信息

URL

> https://platform.armvm.com/distribute/task/query-flow-result

请求格式

> POST

版本号

> 3

使用场景

> 创建流量查询任务后，用返回的taskId调用该接口获取流量查询任务的结果

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务ID	89695

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		返回状态，0-成功	0
data	Object	必须			
├── taskId	Long	必须		任务ID	89695
├── taskDesc	String	必须		创建任务传入的任务描述	"2023年2月份流量计费汇总"
├──billingType	String	必须		计费类型	"FLOW"
├──taskStatus	String	必须		任务状态，INIT 初始化, PROCESSING 处理中, SUCCESS 成功, NOT_EXIST 不存在, FAIL 失败;	"SUCCESS"
├── msg	String	非必须		任务失败时，描述失败原因	
├── billingValue	Double	必须		计费值，成功状态时有值，流量（单位byte）或带宽（单位bps）	100.0
├── bandwidthList	Object[]	非必须		带宽明细列表	
├── ├── recordTime	Long	必须		记录时间（毫秒时间戳）	1676020200000
├── ├── send	Double	必须		发送数据，单位：bps	50.0
├── ├── receive	Double	必须		接收数据，单位：bps	50.0
msg	String	必须		返回信息	
ts	Long	必须		接口返回时时间戳	

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "data": {
    "msg": "",
    "billingValue": 100.0,
    "taskStatus": "SUCCESS",
    "billingType": "FLOW",
    "taskId": "89695",
    "taskDesc": "2023年2月份流量计费汇总",
    "bandwidthList": [
      {
        "recordTime": 1676020200000,
        "receive": 50.0,
        "send": 50.0
      }
    ]
  },
  "ts": "1658455442690"
}
```


异常示例


```
{
  "code": 11108002,
  "msg": "请输入正确的任务ID",
  "ts": 1601192809490
}
```

```
{
  "code": 11100050,
  "msg": "系统异常，请联系客服人员",
  "ts": 1601192809490
}
```


分页查询告警策略

基本信息

URL

> https://platform.armvm.com/merchant/alarm-strategy/page

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
page	Integer	必须		页码，最小1页	1
rows	Integer	必须		每页记录数，最小1条，最大200条	200
alarmStrategyName	String	非必须		告警策略名称，模糊查询	
alarmResourceType	String	非必须		告警资源类型： 1. OSS 对象存储、 2. DUFS DuFS存储、 3. BANDWIDTH 带宽用量	OSS
enableStatus	Integer	非必须		启用状态：0禁用、1启用	0

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├──total	Integer	必须		总数	20
├──totalPage	Integer	必须		总页码	2
├──page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──pageData	Object[]	必须			
├────────alarmStrategyId	Integer	必须		告警策略id	
├────────alarmStrategyName	String	必须		告警策略名称	
├────────alarmStrategyDesc	String	必须		告警策略描述	
├────────idcNames	String[]	必须		机房名称	
├────────alarmResourceType	String	必须		告警资源类型	
├────────alarmMetrics	String	必须		告警维度	
├────────alarmThreshold	Integer	必须		告警阈值，月带宽峰值单位Mps，其余单位为%(百分比)	92Mbps、80%
├────────alarmSilencePeriod	Integer	必须		告警静默间隔；单位分钟	
├────────smsNotifyStatus	Integer	必须		短信通知；0-关闭，1-开启	
├────────smsNotifyObjects	String[]	非必须		短信通知对象，脱敏后的手机号	153****6633
├────────callbackNotifyStatus	Integer	必须		回调通知；0-关闭，1-开启	
├────────enableStatus	Integer	必须		启用状态；0-禁用，1-启用	
├────────createTime	Date	必须		创建时间	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389326710

请求示例

> 参考：调用方式

响应示例


```
{
  "data": {
    "page": 1,
    "rows": 10,
    "totalPage": 1,
    "total": 1,
    "pageData": [
      {
        "alarmStrategyId": "12279594500259850",
        "alarmStrategyName": "告警策略",
        "alarmStrategyDesc": "",
        "idcNames": [
          "广州机房"
        ],
        "alarmResourceType": "DuFS存储",
        "alarmMetrics": "已用量百分比",
        "alarmThreshold": 80,
        "alarmSilencePeriod": 10,
        "smsNotifyStatus": 1,
        "smsNotifyObjects": [
          "158****6464"
        ],
        "callbackNotifyStatus": 1,
        "enableStatus": 1,
        "createTime": 1745581271549
      }
    ]
  },
  "code": 0,
  "msg": "OK",
  "ts": "1750387176775"
}
```


新增告警策略

基本信息

URL

> https://platform.armvm.com/merchant/alarm-strategy/add

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
alarmStrategyName	String	必须		告警策略名称，长度必须为3-64个字符，必须以中英文开头，支持大小写字母，数字和'-_/.'	
alarmStrategyDesc	String	非必须		告警策略描述，最大长度255	
alarmResourceType	String	必须		告警资源类型 1. OSS 对象存储 2. DUFS DuFS存储 3. BANDWIDTH 带宽用量	
idcCodes	String[]	必须		机房编码,最多支持14个； 一个机房可以创建1个带宽用量告警，创建5个对象存储、DuFS存储	
alarmMetrics	String	必须		告警维度 1. remaining-percentage 剩余百分比 单位 % 2. used-percentage 已用量百分比 单位 % 3. bandwidth-peak 当月带宽峰值 单位Mbps	
alarmThreshold	Integer	必须		告警阈值 当月带宽峰值范围[10, 99999] 剩余百分比和已用量百分比范围[1, 99]	
alarmSilencePeriod	Integer	非必须		告警静默间隔,范围[1, 60],单位分钟；对象存储、DUFSDuFS存储必填，带宽用量无需填写	
smsNotifyStatus	Integer	必须		短信通知；0-关闭，1-开启	
smsNotifyObjects	String[]	非必须		短信通知对象，最多支持5个，若开启短信通知，此字段必填； 同一手机号码每分钟最多能接收的短信通知3个，每小时20个，每天能接收的短信通知最多100个	
callbackNotifyStatus	Integer	必须		回调通知；0-关闭，1-开启；**短信通知和回调通知至少开启一个**	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	非必须			
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389326710

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {

  },
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```


异常示例


```
{
  "code": 11101191,
  "msg": "存在相同的告警策略名称",
  "ts": "1750389326710"
}
```


更新告警策略

基本信息

URL

> https://platform.armvm.com/merchant/alarm-strategy/update

请求格式

> POST

版本号

> 3

请求参数

body

只会更新传入字段；若不想更新某个非必填字段，不传对应字段即可

名称	类型	是否必须	默认值	备注	示例值
alarmStrategyId	Long	必须		告警策略id	
alarmStrategyName	String	非必须		告警策略名称，长度必须为3-64个字符，必须以中英文开头，支持大小写字母，数字和'-_/.'	
alarmStrategyDesc	String	非必须		告警策略描述，最大长度255	
alarmThreshold	Integer	非必须		告警阈值 当月带宽峰值范围[10, 99999] 剩余百分比和已用量百分比范围[1, 99]	
alarmSilencePeriod	Integer	非必须		告警静默间隔,范围[1, 60],单位分钟；对象存储、DUFSDuFS存储可以修改，带宽用量不能修改	
smsNotifyStatus	Integer	非必须		短信通知；0-关闭，1-开启	
smsNotifyObjects	String[]	非必须		短信通知对象，填写手机号码，最多支持5个，若开启短信通知，此字段必填； 同一手机号码每分钟最多能接收的短信通知3个，每小时20个，每天能接收的短信通知最多100个	
callbackNotifyStatus	Integer	非必须		回调通知；0-关闭，1-开启；**短信通知和回调通知至少开启一个**	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	非必须			
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389326710

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {

  },
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```

异常示例

```
{
  "code": 11101194,
  "msg": "告警策略不存在",
  "ts": "1750389326710"
}
```


删除告警策略

基本信息

URL

> https://platform.armvm.com/merchant/alarm-strategy/remove

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
alarmStrategyIds	Long[]	必须		告警策略id,最大1000个	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	非必须			
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389268764

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {

  },
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```


异常示例


```
{
  "code": 11100003,
  "msg": "参数请求错误，告警策略ID不能为空！",
  "ts": "1750389268764"
}
```


启/禁用告警策略

基本信息

URL

> https://platform.armvm.com/merchant/alarm-strategy/update-enable-status

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
alarmStrategyIds	Long[]	必须		告警策略id,最大1000个	
enableStatus	Integer	必须		启用状态，0-禁用，1-启用	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	非必须			
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1750389268764

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {

  },
  "code": 0,
  "msg": "OK",
  "ts": "1750389326710"
}
```

异常示例

```
{
  "code": 11100003,
  "msg": "参数请求错误，告警策略ID不能为空！",
  "ts": "1750389268764"
}
```

任务管理

获取任务类型列表

基本信息

URL

> https://platform.armvm.com/command/pad/execute-task-type-list

请求格式

> POST

版本号

> 3

接口说明

> 用于获取任务类型列表

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├──type	String	必须		任务类型	
├──name	Long	必须		任务名称	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1762311419902

请求示例

> 参考：调用方式

响应示例

```
{
  "msg": "OK",
  "code": 0,
  "data": [
    {
      "type": "uploadApp",
      "name": "应用同步"
    },
    {
      "type": "installApp",
      "name": "应用安装"
    }
  ],
  "ts": 1762311419902
}
```

分页查询任务信息

基本信息

URL

> https://platform.armvm.com/command/pad/execute-task-page

请求格式

> POST

版本号

> 3

接口说明

> 用于分页查询任务结果，taskIds与taskType参数不能同时为空；若同时传入两个参数，结果需同时满足两个条件；

> taskType的值请看获取任务类型列表文档

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long[]	非必须		任务ID列表，最多200个	[1001,1002]
taskType	String	非必须		任务类型	"instanceReboot"
page	Integer	必须		页码，最小为1	1
rows	Integer	必须		每页记录数，最小1条，最大200条	20

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├──page	Integer	必须		页码	
├──rows	Integer	必须		每页记录数	
├──totalPage	Integer	必须		总页数	
├──total	Integer	必须		总记录数	
├──pageData	Object[]	非必须			
├────taskId	Long	必须		任务ID	1000
├────padCode	String	必须		实例编号	VM192168169028
├────taskStatus	Integer	必须		任务状态， 5-待执行 6-执行中 7-执行成功 8-执行失败	7
├────taskResult	String	必须		任务结果	success
├────taskType	String	必须		任务类型	
├────createTime	Long	必须		任务创建时间	
├────executeTime	Long	必须		任务执行完成时间	
├────deviceIp	String	必须		物理机IP	
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "msg": "OK",
  "code": 0,
  "data": {
    "page": 1,
    "rows": 10,
    "totalPage": 2,
    "total": 13,
    "pagedata": [
      {
        "padCode": "VM011010067010",
        "taskId": 2813218757970419736,
        "taskStatus": 7,
        "taskResult": "success",
        "deviceIp": "192.168.1.1",
        "taskType": "instanceReboot",
        "createTime": 1772820359299,
        "executeTime": 1772820359299
      }
    ]
  },
  "ts": 1762311419902
}
```


异常示例


```
{
  "code": 11100003,
  "msg": "参数请求错误，taskIds与taskType不能同时为空",
  "ts": 1762311419902
}
```


任务结果查询

基本信息

URL

> https://platform.armvm.com/command/pad/execute-task-info.html

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long[]	必须		任务ID列表，最多200个	[1001,1002]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	1000
├─padCode	String	必须		实例编号	VM192168169028
├─taskStatus	Integer	必须		任务状态， 5-待执行 6-执行中 7-执行成功 8-执行失败	7
├─taskResult	String	必须		任务结果	success
├─taskType	String	必须		任务类型	
├─createTime	Long	必须		任务创建时间	
├─executeTime	Long	必须		任务执行完成时间	
├─deviceIp	String	必须		物理机IP	
├─creater	String	非必须		任务发起人	demo_user
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例


```
{
  "msg": "OK",
  "code": 0,
  "data": [
    {
      "padCode": "VM011010067010",
      "taskId": 2813218757970419736,
      "taskStatus": 7,
      "taskResult": "success",
      "deviceIp": "192.168.1.1",
      "taskType": "instanceReboot",
      "createTime": 1772820359299,
      "executeTime": 1772820359299,
      "creater": "demo_user"
    }
  ],
  "ts": 1762311419902
}
```


异常示例

```
{
  "code": 11108002,
  "msg": "请输入正确的任务ID",
  "ts": 1672820359299
}
```


终止任务

基本信息

URL

> https://platform.armvm.com/resources/instance/event-task-stop

请求格式

> POST

版本号

> 3

使用场景

> 支持对状态为“待执行”的任务发起终止

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long[]	必须		任务ID列表，最大200个	[12244837858185221]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1744275100700

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "",
  "ts": 1744275100700
}
```


异常示例


```
{
"code": 11108041,
"msg": "工作流信息异常",
"ts": 1744275100700
}
```

```
{
"code": 11108043,
"msg": "只允许终止待执行的工作流任务",
"ts": 1744275100700
}
```


系统镜像

查询公共镜像

基本信息

URL

> https://platform.armvm.com/resources/instance-base-image/list

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceServerType	String	非必须		实例服务器类型	3588
romVersion	String	非必须		系统版本： android6.0 android8.1 android10.0 android12.0 android13.0	android8.1
imageVersionIds	Long[]	非必须		镜像版本id，最多200个	[5]
page	Integer	必须		页码，最小值为1	1
rows	Integer	必须		每页条数，最小值为1，最大值为200	200

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├── total	Integer	必须		总数	20
├── totalPage	Integer	必须		总页码	2
├── page	Integer	必须		当前页码	1
├── rows	Integer	必须		每页数量	10
├── pageData	Object[]	非必须			
├──── imageVersionId	Long	必须		镜像版本ID	5
├──── imageVersionName	Long	必须		镜像版本名称	5
├──── instanceServerType	String	必须		实例服务器类型	3588
├──── romVersion	String	必须		系统版本	android8.1
├──── createTime	Long	必须		上传时间（时间戳）	1601188620078
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "total": 1,
    "totalPage": 1,
    "page": 1,
    "rows": 10,
    "pageData": [
      {
        "imageVersionId": "5",
        "imageVersionName": "baidu_phone_845_aosp8.1_v2.4.0_20240130172159.tar.gz",
        "instanceServerType": "3588",
        "romVersion": "android8.1",
        "createTime": 1716819459335
      }
    ]
  },
  "msg": "",
  "ts": 1601021167163
}
```


查询自定义镜像

基本信息

URL

> https://platform.armvm.com/resources/instance-image/list

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
page	Integer	非必须	1	页码，最小值为1	1
rows	Integer	非必须	200	每页条数，最小值为1，最大值为200	200
imageVersionId	Long	非必须		镜像版本id	10001
imageVersionIds	Long[]	非必须		镜像版本id，最多200个	[10001,10002]
instanceServerType	String	非必须		实例服务器类型	3588
romVersion	String	非必须		系统版本： android6.0 android8.1 android10.0 android12.0 android13.0	android8.1
imageVersionName	String	非必须		镜像版本名称，模糊查询	xxx

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├── total	Integer	必须		总数	20
├── totalPage	Integer	必须		总页码	2
├── page	Integer	必须		当前页码	1
├── rows	Integer	必须		每页数量	10
├── pageData	Object[]	非必须			
├──── imageVersionId	Long	必须		镜像版本ID	10001
├──── imageUploadStatus	String	必须		镜像上传状态，wait_execute：待执行、executing：执行中、success：成功、fail：失败	success
├──── instanceServerType	String	必须		实例服务器类型	3588
├──── romVersion	String	必须		系统版本	android8.1
├──── imageFiles	Object[]	必须		镜像文件信息	
├──── imageFileUrl	String	必须		镜像文件下载地址	http://xxx.img
├──── imageFileName	String	必须		镜像文件名称（带后缀）	root_aosp10_xx.img
├──── imageFileType	String	必须		odm_aosp root_aosp system_aosp vendor_aosp super_aosp	root_aosp
├──── imageFileMd5	String	必须		镜像文件md5值	xxxxxxx
├──── baseImageVersionId	Long	必须		镜像底包ID	5
├──── imageVersionName	String	必须		镜像版本名称	xxx
├──── describe	String	必须		镜像版本描述	xxx
├──── createTime	Long	必须		上传时间（时间戳）	1601188620078
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例


```
{
  "code": 0,
  "data": {
    "total": 1,
    "totalPage": 1,
    "page": 1,
    "rows": 10,
    "pageData": [
      {
        "imageVersionId": 10001,
        "imageUploadStatus": "success",
        "instanceServerType": "3588",
        "romVersion": "android8.1",
        "imageFiles": [
          {
            "imageFileUrl": "https://baidu.com/armvm/root-v15.img",
            "imageFileName": "root_aosp-v15.img",
            "imageFileType": "root_aosp",
            "imageFileMd5": "20d41ecb84d86667525a8a02d0102f2b"
          },
          {
            "imageFileUrl": "https://baidu.com/armvm/system-v15.img",
            "imageFileName": "system_aosp-v15.img",
            "imageFileType": "system_aosp",
            "imageFileMd5": "22aca472f2d66551f197ff14de8618f3"
          }
        ],
        "baseImageVersionId": 5,
        "imageVersionName": "xxx",
        "describe": "xxx",
        "createTime": 1601021167163
      }
    ]
  },
  "msg": "",
  "ts": 1601021167163
}
```


上传自定义镜像

基本信息

URL

> https://platform.armvm.com/resources/instance-image/upload

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
imageFiles	Object[]	必须		镜像文件信息，最多20个	
├─ imageFileUrl	String	必须		镜像文件下载地址	http://xxx.img
├─ imageFileName	String	必须		镜像文件名称。 镜像文件名称需要以镜像文件类型为前缀，比如说root_aosp类型的就需要以root_aosp开头命名，如root_aosp_xx.img ； 如果是安卓12，在文件类型后面应该带f2fs，如system_aosp_f2fs_xxx.img	root_aosp10_xx.img
├─ imageFileType	String	必须		odm_aosp root_aosp system_aosp vendor_aosp super_aosp	root_aosp
├─ imageFileMd5	String	必须		镜像文件md5值	xxxxxxx
instanceServerType	String	必须		实例服务器类型	3588
romVersion	String	必须		系统版本： android6.0 android8.1 android10.0 android12.0 android13.0	android8.1
baseImageVersionId	Long	非必须		镜像底包ID	5
imageVersionName	String	必须		镜像版本名称，最多50个字符	xxx
describe	String	非必须		镜像版本描述，最多80个字符	xxx

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	必须			
├─ imageVersionId	Long	必须		镜像版本ID	10001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "imageVersionId": 10001
  },
  "msg": "",
  "ts": 1601021167163
}
```

获取自定义镜像上传结果

基本信息

URL

> https://platform.armvm.com/resources/instance-image/upload-info

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
imageVersionId	Long	必须		实例镜像版本ID	10001

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	必须			
├── imageUploadStatus	String	必须		镜像上传状态，wait_execute|executing:执行中|success:成功|fail:失败	success
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
     "imageUploadStatus": "success"
  },
  "msg": "",
  "ts": 1601021167163
}
```

删除自定义镜像

基本信息

URL

> https://platform.armvm.com/resources/instance-image/remove

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
imageVersionIds	Long[]	必须		实例镜像版本ID	[10001]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	必须			
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1728627866628

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {},
  "msg": "",
  "ts": 1728627866628
}
```


创建实例更新镜像任务

基本信息

URL

> https://platform.armvm.com/resources/instance-image/update

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
imageVersionId	Long	必须		镜像版本id	5
instanceCodes	String[]	必须		实例编号集合，最多200个	["VM010100240100"]
configId	String	否		配置项ID	
resourcePackageId	Long	否		资源包ID	
reset	Bool	否	false	是否恢复出厂设置	false
autoInstall	Bool	否	true	是否自动安装镜像	true

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object []	必须			
├── instanceCode	String	必须		实例编号	VM010100240100
├── taskId	Long	必须		任务id	9001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "instanceCode": "VM0101002401OO",
      "taskId": 9001
    }
  ],
  "msg": "",
  "ts": 1601021167163
}
```

查询实例更新镜像结果

基本信息

URL

> https://platform.armvm.com/resources/instance-image/update-info

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long[]	必须		任务ID集合，最多200个	[8001,8002]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├─taskId	Long	必须		任务ID	8001
├─ instanceCode	String	必须		实例编号	VM010100240100
├─ imageUpdateStatus	String	必须		镜像更新状态，wait_execute:待执行|executing:执行中 success:成功 | fail:失败	success
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": [
  {
   "taskId": 8001,
   "instanceCode": "VM010100240100",
   "imageUpdateStatus": "success"
  },
  {
   "taskId": 8002,
   "instanceCode": "VM010100240101",
   "imageUpdateStatus": "executing"
  }
 ],
 "msg": "",
 "ts": 1601021167163
}
```

创建物理机更新镜像任务

注意事项

1.整个物理机下的所有实例均会清除镜像内数据且不可恢复，请谨慎操作。更新镜像过程中，云手机服务会中断几分钟。
2.若系统版本由高到低的降级操作（例如从 android 12.0 降至 android 10.0），需恢复出厂设置后才可正常使用。
3.需重启物理机（不重启内核不生效，可能导致启动异常）

基本信息

URL

> https://platform.armvm.com/resources/device/image-update

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
imageVersionId	Integer	是		镜像版本id	10001
deviceIps	String[]	是		物理机IP集合，最多200个	["10.10.240.2"]
configId	String	否		配置项ID	
resourcePackageId	Long	否		资源包ID	
reset	Bool	否	false	是否恢复出厂设置	false
autoInstall	Bool	否	true	是否自动安装镜像	true

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├──deviceIp	String	是		物理机IP	10.10.240.2
├── taskId	Long	必须		任务id	9001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "deviceIp": "10.10.240.2",
      "taskId": 9001
    }
  ],
  "msg": "",
  "ts": 1601021167163
}
```

物理机管理

分页查询物理机信息

基本信息

URL

> https://platform.armvm.com/resources/device/page

请求格式

> POST

版本号

> 3

使用场景

> 根据查询条件分页获取物理机列表信息。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
page	Integer	非必须	1	页码，默认值1，最小1页（包含1页）	1
rows	Integer	非必须	200	每页记录数，默认值200，最小1条（包含1条），最大1000条（包含1000条）	10
deviceCodes	String[]	非必须		物理机编号集合，集合长度在200内（包含200）	["GZNX-IDC_DM011010099026"]
deviceIps	String[]	非必须		物理机IP集合，集合长度在200内（包含200）	["192.168.1.1"]
deviceIpSegment	String	非必须		物理机IP段	192.168.1
idcCode	String	非必须		所属机房编号	GZNX-IDC-02
instanceServerType	String	非必须		服务器类型	3588
romVersion	String	非必须		系统版本	android12.0
deviceStatus	Integer	非必须		物理机在线状态：0:离线；1:在线	1
includeSubMerchant	Boolean	非必须	false	是否包含子商户，为true时将查询子商户的物理机，如果物理机属于多个商户会聚合成一条记录	false

返回参数

名称	类型	是否必须	备注	示例值
code	Integer	必须	响应编码	0
data	Object	非必须		
├──total	Integer	必须	总数	20
├──totalPage	Integer	必须	总页码	2
├──page	Integer	必须	页码	1
├──rows	Integer	必须	每页数量	10
├──pageData	Object []	必须		
├────────deviceIp	String	必须	物理机IP	192.168.1.100
├────────deviceIpSegment	String	必须	物理机IP段	192.168.1
├────────deviceCode	String	必须	物理机编号	GZNX-IDC_DM011010099026
├────────idcCode	String	必须	机房编号	GZNX-IDC-02
├────────idcName	String	必须	机房名称	南翔机房-可用区1
├────────instanceServerType	String	必须	服务器类型	3588
├────────romVersion	String	必须	系统版本	android12.0
├────────instanceGrade	Integer	必须	开数	1
├────────instanceGradeLimit	Integer	必须	多开上限	5
├────────allocateTime	Date	必须	调度时间（时间戳）	1733297481137
├────────recycleTime	Date	必须	回收时间（时间戳）	1733297481137
├────────memoryModel	String	必须	内存规格	16GB
├────────cpuModel	String	必须	CPU规格	8C
├────────diskModel	String	必须	磁盘规格	256GB
├────────deviceStatus	Integer	必须	物理机在线状态：0:离线；1:在线	1
├────────merchantNames	String	必须	商户名称集合，多个商户名称使用、分割	商户A、商户B
├────────currentMerchantName	String	必须	当前商户名称	商户A
├────────deviceUpTime	Long	必须	物理机运行时长，单位秒	23423
msg	String	必须	响应消息	
ts	Long	必须	时间戳	1733297481137

请求示例

> 参考：调用方式

响应示例

```
{
 "code": 0,
 "data": {
  "total": 20,
  "totalPage": 2,
  "page": 1,
  "rows": 10,
  "pageData": [
   {
    "deviceIp": "192.168.1.100",
    "deviceIpSegment": "192.168.1",
    "deviceCode": "GZNX-IDC_DM011010099026",
    "idcCode": "GZNX-IDC-02",
    "idcName": "南翔机房-可用区1",
    "instanceServerType": "3588",
    "romVersion": "android12.0",
    "instanceGrade": 1,
    "instanceGradeLimit": 5,
    "allocateTime": 1733297481137,
    "recycleTime": 1733297481137,
    "memoryModel": "16GB",
    "cpuModel": "8C",
    "diskModel": "256GB",
    "deviceStatus": 1,
    "merchantNames": "商户A、商户B",
    "currentMerchantName": "商户A",
    "deviceUpTime": 23423
   }
  ]
 },
 "msg": "success",
 "ts": 1733297481137
}
```


重启物理机

基本信息

URL

> https://platform.armvm.com/resources/device/reboot

请求格式

> POST

版本号

> 3

使用场景

> 重启指定物理机

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
deviceCodes	String[]	必须		物理机编号，最多200个	["CS-IDC-01_DM001000000001"]
rebootType	Integer	非必须	0	重启类型；0-软重启，1-硬重启，2-断电重启	1

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务id	12259080139067404
├─deviceCode	String	必须		物理机编号	"CS-IDC-01_DM001000000001"
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1747035069000

请求示例

> 参考：调用方式

响应示例

```
{
"data": [
    {
        "taskId": 12259080139067404,
        "deviceCode": "CS-IDC-01_DM001000000001"
    }
],
"code": 0,
"msg": "OK",
"ts": 1747035069000
}
```

异常示例

```
{
"code": 11103250,
"msg": "物理机不存在",
"ts": 1747035069000
}
```

商户管理

新增管理员账号

基本信息

URL

> https://platform.armvm.com/merchant/open-account/save

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
userName	String	必须		管理员用户名，由数字、字母和-、_组成，最长16个字符	admin
phone	String	必须		管理员手机号码	
nickname	String	必须		管理员昵称	admin
roleNames	String[]	非必须		关联角色名称集合，填写前需到后台-账号管理-角色管理中新增角色，集合长度最大为200	["test"]
merchantPoolNos	Long[]	非必须		关联实例池编号集合，集合长度最大为200	[10000]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─password	String	必须		管理员密码	
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {
    "password": "xxxxx"
  },
  "code": 0,
  "msg": "OK",
  "ts": "1750231776858"
}
```

异常示例

```
{
  "code": 11107008,
  "msg": "角色[xxx]不存在",
  "ts": "1750231743378"
}
```

新增子商户

基本信息

URL

> https://platform.armvm.com/merchant/sub-merchant/add

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
parentMerchantCode	String	非必须		父商户编码，如果为空则代表在当前商户下新增子商户	10000
merchantCode	String	必须		商户编码，由字母、数字、_和-组成，最长32个字符	10001
merchantName	String	必须		商户名称，最长32个字符	test
merchantType	String	必须		商户类型，cloud_app-云游戏，cloud_phone-云手机，OEM-速享版，支持选择多个，多个以英文逗号(,)隔开	cloud_app,cloud_phone
merchantPhone	String	非必须		商户联系人手机号码	
adminUserName	String	必须		超级管理员用户名，由数字、字母和-、_组成，最长16个字符	admin
adminPhone	String	必须		超级管理员手机号码	

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─adminPassword	String	必须		超级管理员密码	
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "data": {
    "adminPassword": "xxxxx"
  },
  "code": 0,
  "msg": "OK",
  "ts": "1750231776858"
}
```

异常示例

```
{
  "code": 11101076,
  "msg": "商户不存在",
  "ts": "1750231743378"
}
```

机型管理

分页查询机型信息

基本信息

URL

> https://platform.armvm.com/resources/template-model-page

请求格式

> POST

版本号

> 3

使用场景

> 根据查询条件分页获取机型列表信息。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
page	Integer	必须		页码，最小1页	1
rows	Integer	必须		每页记录数，最小1条，最大200条	10
brand	String	非必须		品牌	HUAWEI
manufacturer	String	非必须		制造商	HUAWEI
model	String	非必须		型号	LIO-AN00
enableStatus	Integer	非必须		启用状态：0-禁用，1-启用	1

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──total	Integer	必须		总数	20
├──totalPage	Integer	必须		总页码	2
├──page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──pageData	Object []	必须			
├────templateModelId	Long	必须		模板机型ID	11936974100295684
├────brand	String	必须		品牌	HUAWEI
├────manufacturer	String	必须		制造商	HUAWEI
├────model	String	必须		型号	LIO-AN00
├────enableStatus	String	必须		启用状态：0-禁用，1-启用	0
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "total": 1,
    "totalPage": 1,
    "page": 1,
    "rows": 10,
    "pageData": [
      {
        "templateModelId": 11936974100295684,
        "brand": "HUAWEI",
        "manufacturer": "HUAWEI",
        "model": "LIO-AN00",
        "enableStatus": 1
      }
    ]
  },
  "msg": "",
  "ts": 1598754678123
}
```


修改机型启用状态

基本信息

URL

> https://platform.armvm.com/resources/template-model-enable-status

请求格式

> POST

版本号

> 3

使用场景

> 修改机型启用状态

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
templateModelIds	Long[]	必须		模板机型ID列表，最大200个	[11936974100295684]
enableStatus	Integer	必须		启用状态：0-禁用，1-启用	1

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "",
  "ts": 1598754678123
}
```

备份还原

实例备份

基本信息

> 对云手机实例进行数据备份，生成一份快照数据保存到对象存储中，之后可以通过实例还原功能将快照数据恢复到云手机实例中。
> **备份还原会产生大量流量，请谨慎使用外网的对象存储服务。**

URL

> https://platform.armvm.com/resources/instance/backup

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCode	String	必须		实例编号	VM192168001001
snapshotName	String	必须		快照名称	VM192168001001_0925
ossConfig	Object	必须		对象存储配置，需要支持s3协议	
├──endpoint	String	必须		对象存储endpoint	s3.bj.bcebos.com
├──bucket	String	必须		对象存储bucket	test
├──accessKey	String	必须		对象存储accessKey	ATqjcosksxz
├──secretKey	String	必须		对象存储secretKey	sdfaweradfg
├──protocol	String	必须		对象存储访问协议：http, https	https
snapshotPath	String	非必须	/	快照数据保存在对象存储的路径，不传则默认保存在根目录"/"下	/snapshot/data/
excludes	String[]	非必须		要排除的目录(文件)	["example.txt"]
includes	String[]	非必须		要包含的目录(文件)	["example.txt"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object	必须			
├──taskId	Long	必须		任务ID	100000
├──instanceCode	String	必须		实例编号	VM192168001001
├──snapshotId	Long	必须		快照id	7290017192180314133
├──snapshotName	String	必须		快照名称	VM192168001001_0925
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "taskId": 100000,
    "instanceCode": "VM192168001001",
    "snapshotId": 7290017192180314133,
    "snapshotName": "VM192168001001_0925"
  },
  "msg": "",
  "ts": 1601188620078
}
```


异常示例


```
{
 "code": 11103062,
 "msg": "实例不存在",
 "ts": 1601021167163
}
```


实例还原

基本信息

> 将已备份成功的快照数据恢复到指定的云手机实例。
> **备份还原会产生大量流量，请谨慎使用外网的对象存储服务。**

URL

> https://platform.armvm.com/resources/instance/restore

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
snapshotId	Long	必须		快照ID	7290017192180314133
instanceCodes	String[]	必须		实例编号集合，最多200个	["VM192168001001"]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	必须			
├─taskId	Long	必须		任务ID	100000
├─instanceCode	String	必须		实例编号	VM192168001001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM192168001001"
     }
  ],
  "msg": "",
  "ts": 1601188620078
}
```

异常示例

```
{
 "code": 11108050,
 "msg": "实例快照未完成创建",
 "ts": 1601021167163
}
```

实例备份删除

基本信息

> 删除实例备份的快照数据。

URL

> https://platform.armvm.com/resources/instance/snapshot-remove

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
snapshotIds	Long[]	必须		快照ID集合，最多200个	[7290017192180314133]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
msg	String	必须		错误信息	"OK"
ts	Long	必须		时间戳	1768190400205

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "msg": "OK",
  "ts": 1768190400205
}
```

异常示例

```
{
 "code": 11108053,
 "msg": "快照正在制作中，暂不允许被删除",
 "ts": 1768190400205
}
```

分页查询实例备份信息

基本信息

URL

> https://platform.armvm.com/resources/instance/snapshot-page

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
snapshotIds	Long[]	非必须		快照ID集合	[72900171921803141 33]
snapshotName	String	非必须		快照名称，可模糊查询	
snapshotStatus	String	非必须		快照状态： wait_create-待创建 creating-创建中 create_success-创建成功 create_failed-创建失败	create_success
page	Integer	必须		页码	1
rows	Integer	必须		每页条数，最多200	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──total	Integer	必须		总数	20
├──totalPage	Integer	必须		总页码	2
├──page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──pageData	Object []	必须			
├────────snapshotId	Long	必须		快照ID	72900171921803141 33
├────────snapshotName	String	必须		快照名称	VM192168001001_0925
├────────snapshotStatus	String	必须		快照状态： wait_create-待创建 creating-创建中 create_success-创建成功 create_failed-创建失败	create_success
├────────snapshotSize	Integer	必须		快照大小，单位KB	2048
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": {
    "total": 1,
    "totalPage": 1,
    "page": 1,
    "rows": 10,
    "pageData": [
      {
        "snapshotId": 7290017192180314133,
        "snapshotName": "VM192168001001_0925",
        "snapshotStatus": "create_success",
        "snapshotSize": 2048
      }
    ]
  },
  "msg": "",
  "ts": 1601021167163
}
```


异常示例

快照能力

快照接口使用说明

快照定义

> 快照是存储在机房对象存储服务中的实例 /data 卷的完整备份。快照包含 /data 卷上的所有数据，例如已安装的应用、应用生成的数据以及文件数据等。需要注意的是，由于快照仅支持备份 /data 卷的数据，因此无法使用此功能备份和还原存储在系统卷中的数据，例如预置的应用等。

使用前提

> 如果您需要使用快照能力，请与PMO联系，需要配置一个合理的限额，此外，因设备需要做特殊处理，请确保您的设备符合使用要求。

注意事项

快照备份要求

> 要将实例的数据备份为一个快照，必须先经历一次初始化或挂载流程，否则无法备份。

快照删除限制

> 正在创建或待创建的快照不允许被删除；已被实例挂载的快照也不允许删除。

跨服务器和安卓版本支持

> 系统支持跨服务器类型进行快照挂载，然而，在跨安卓版本方面，仅支持低版本安卓实例生成的快照挂载到高版本安卓实例上。例如，系统版本为安卓 10 的实例所制作的快照可以挂载到安卓 12 的实例上，但安卓 12 实例制作的快照无法挂载到安卓 10 实例上。为了保证挂载的稳定，建议在相同服务器类型和系统版本之间使用快照功能。

快照重新挂载

为了确保业务的灵活性，允许已挂载快照的实例重新挂载一个新的快照。然而，请注意以下事项：如果在第一次挂载快照后，实例产生了增量数据，必须先卸载当前快照（即创建一个新的快照以保存数据），然后再挂载其他快照。如果不先进行卸载而直接挂载新的快照，实例在第一次挂载后产生的增量数据将会被清空。

功能列表

初始化

用途：在需要创建 0-1 快照的情况下，实例必须先进行初始化。未初始化的设备所生成的数据将无法进行备份。此外，初始化还有一个重要用途：当快照功能出现异常（例如挂载失败或卸载失败等情况）时，通过执行初始化操作，可以放弃当前数据，使实例恢复到可用状态。 初始化的流程：初始化的过程包括清空实例当前的所有数据，并挂载一个空目录。因此，初始化可以视为一种特殊的挂载操作，其挂载的是一个空快照。

挂载

用途：挂载是将对象存储中的远程快照文件挂载到实例上，其实际效果是还原数据到实例。 挂载的流程：将本地的数据清理，挂载一个远程快照，将远程快照中部分文件读取到本地后，启动实例。

卸载

用途：卸载是将实例当前的 `/data` 卷备份到机房对象存储中，并生成一个快照。其实际效果是创建一个对象存储文件，从而实现数据的备份。 卸载的流程：将当前data卷数据文件备份到对象存储，产生一个快照，并将数据从实例内清空，变回一个空的状态。

实例相关字段说明

挂载快照状态: 当前实例是否挂载快照 挂载快照ID: 当前正在挂载的快照编号

一次完整的快照使用流程

1、选择实例：选择一个处于空状态的实例，用于 0-1 快照制作。 2、初始化：将选择的实例进行初始化，并等待初始化任务完成。 3、使用实例：实例初始化完成后，在实例的 `/data` 卷中产生数据，例如安装应用程序等。 4、卸载快照：使用完成后，对该实例进行快照卸载，等待卸载完成，实例将恢复为未挂载状态。 5、查询快照：查询快照列表，找到刚刚生成的快照，并复制其快照 ID。 6、选择实例：找到一个或多个处于空状态的实例，用于挂载已创建的快照。 7、挂载快照：对选择的实例执行挂载快照的操作，并等待挂载任务完成。 8、使用实例并保存数据：在已挂载快照的实例上产生增量数据，然后再次卸载以保存数据。

快照各流程中云手机的相关状态

空状态

挂载快照状态: 未挂载 挂载快照ID: 无

初始化完成

> 挂载快照状态: 已挂载 挂载快照ID: 无

挂载完成

> 挂载快照状态: 已挂载 挂载快照ID: 有

卸载完成

> 挂载快照状态: 未挂载 挂载快照ID: 无

常见问题

1、**挂载/卸载 和 备份/还原 的区别？**

> 挂载/卸载 比 备份/还原 时效性更高。备份/还原 是需要把全量备份数据拷贝到本地后才能开机；挂载/卸载 是只需将分片数据中某些数据拷贝到本地后即可开机。

2、挂载快照后实例会被恢复到什么程度？

> 只会把数据还原到 /data卷，但实际登录状态，使用状态是没有保留的。

3、快照使用的存储空间是如何计算的？

> 卸载操作所产生的快照，会保存到对象存储里占用存储空间，且相同文件会重复计算空间。

> 例如某快照大小为100MB，此时需要计费的存储空间为100MB，不管被多少数量的实例挂载，计费的存储空间均为100MB；如果有十个实例挂载此快照，产生使用数据后也需要保存每个实例的快照，比如每台实例都在原本快照的基础上产生了额外30MB的数据，那么10个实例的快照卸载完成后，总的快照空间大小应为：100MB + (100MB + 30MB) x 10 = 1400MB。

快照初始化

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/init

请求格式

> POST

版本号

> 3

使用场景

> 快照初始化

请求参数

名称	类型	是否必须	备注	示例值
instanceCodes	String[]	是	实例编号集合，最多200个	["VM011101001096", "VM011101001097"]
quotaCapacity	Integer	是	云手机的 data 目录可用容量，单位GB。范围： 最小值1GB，最大值1024GB	256
memoryLimit	Integer	否	内存限额大小，单位MB。范围：最小值512MB	1024

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object[]	是			
├─instanceCode	String	是		实例编号	
├─taskId	Long	是		任务ID	
msg	String	是		响应消息	
ts	Long	是		时间戳	1598754678123

请求示例

> 参考：调用方式

快照卸载

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/unmount

请求格式

> POST

版本号

> 3

使用场景

> 对已挂载的实例发起快照卸载，会生成一个新的快照记录

请求参数

名称	类型	是否必须	备注	示例值
instanceCodes	String[]	是	实例编号集合，最多200个	["VM011101001096", "VM011101001097"]
snapshotName	String	是	快照名称	"Test-Snapshot"
online	Boolean	否	是否创建在线快照（空默认 false）	false

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object[]	是			
├─instanceCode	String	是		实例编号	
├─taskId	Long	是		任务ID	
├─snapshotId	Long	是		快照ID	
msg	String	是		响应消息	
ts	Long	是		时间戳	1598754678123

请求示例

> 参考：调用方式

快照挂载

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/mount

请求格式

> POST

版本号

> 3

使用场景

> 挂载已卸载成功产生的快照记录到云手机上

请求参数

名称	类型	是否必须	备注	示例值
instanceCodes	String[]	是	实例编号集合，最多200个	["VM011101001096", "VM011101001097"]
snapshotId	Long	是	快照ID	
quotaCapacity	Integer	否	云手机的 data 目录可用容量，单位GB。范围： 最小值1GB，最大值1024GB	256
memoryLimit	Integer	否	内存限额大小，单位MB。范围：最小值512MB	1024

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object[]	是			
├─instanceCode	String	是		实例编号	
├─taskId	Long	是		任务ID	
msg	String	是		响应消息	
ts	Long	是		时间戳	1598754678123

请求示例

> 参考：调用方式

分页查询快照记录

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/page

请求格式

> POST

版本号

> 3

使用场景

> 分页查询快照记录

请求参数

名称	类型	是否必须	备注	示例值
snapshotName	String	否	快照名称，支持模糊搜索	["TEST-IDC-01_DM011101001096", "TEST-IDC-01_DM011101001097"]
snapshotStatus	String	否	快照状态：待创建-wait_create；创建中-creating；创建成功-create_success；创建失败-create_failed	"create_success"
snapshotIds	Long[]	否	快照ID集合	
page	Integer	是	页码	1
rows	Integer	是	每页条数，最多200	200

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object	是			
├─total	Integer	是		总数	
├─totalPage	Integer	是		总页码	
├─page	Integer	是		页码	
├─rows	Integer	是		每页数量	
├─pageData	Object[]	是			
├──snapshotId	Long	是		快照ID	
├──snapshotName	String	是		快照名称	
├──snapshotStatus	String	是		快照状态	
├──instanceCode	String	是		创建该快照的实例编号	
├──idcCode	String	是		快照所属机房编号	
├──progress	String	是		快照进度	
├──snapshotSize	String	是		快照大小，单位KB	
├──failReason	String	是		失败原因	
msg	String	是		响应消息	
ts	Long	是		时间戳	1598754678123

请求示例

> 参考：调用方式

删除快照

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/remove

请求格式

> POST

版本号

> 3

使用场景

> 删除掉不再需要的快照，只能删除创建成功或创建失败的快照，创建中的快照不允许被删除

请求参数

名称	类型	是否必须	备注	示例值
snapshotIds	Long[]	是	快照ID集合，最多200个	[12214423066714199, 12214423066714198]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object	是			
msg	String	是		响应消息	
ts	Long	是		时间戳	1598754678123

请求示例

> 参考：调用方式

快照初始化任务结果通知

基本信息

URL

> http://host:port/api/v1/instance/dufs-snapshot-init-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用快照初始化接口后，任务结果会通过该回调接口通知给客户。

请求参数

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123123123123
msg	String	必须		返回信息	ok
result	boolean	必须		任务结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

快照卸载任务结果通知

基本信息

URL

> http://host:port/api/v1/instance/dufs-snapshot-unmount-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用快照卸载接口后，任务结果会通过该回调接口通知给客户。

请求参数

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123123123123
msg	String	必须		返回信息	ok
result	boolean	必须		任务结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

快照挂载任务结果通知

基本信息

URL

> http://host:port/api/v1/instance/dufs-snapshot-mount-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用快照挂载接口后，任务结果会通过该回调接口通知给客户。

请求参数

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123123123123
msg	String	必须		返回信息	ok
result	boolean	必须		任务结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

快照批量挂载

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/batch-mount

请求格式

> POST

版本号

> 3

使用场景

> 批量挂载已卸载成功产生的快照记录到云手机上

请求参数

名称	类型	是否必须	备注	示例值
snapshotMountInfos	Object[]	是	最大200个	
├── instanceCode	String	是	实例编码	VM011101001096
├── snapshotId	Long	是	快照ID	
├── quotaCapacity	Integer	否	云手机的 data 目录可用容量，单位GB。最小值1GB，最大值1024GB	256
├── memoryLimit	Integer	否	内存限额大小，单位MB。范围：最小值512MB	1024

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object[]	是			
├──instanceCode	String	是		实例编号	
├──taskId	Long	是		任务ID	
msg	String	是		响应消息	
ts	Long	是		时间戳	1745494806123

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data":[
     {
        "taskId": 100000,
        "instanceCode": "VM011101001096"
     }
  ],
  "msg": "",
  "ts": 1745494806123
}
```


异常示例


```
{
  "code": 11108051,
  "msg": "快照不存在",
  "ts": 1745494806123
}
```


快照配额设置

基本信息

URL

> https://platform.armvm.com/resources/dufs-snapshot/quota-set

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	备注	示例值
instanceCodes	String[]	是	最大200个	["VM192168043080"]
quotaCapacity	Integer	是	快照配额容量，单位GB。范围： 最小值1GB，最大值1024GB，只能向上调整	512

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	是		响应编码	0
data	Object[]	是			
├─instanceCode	String	是		实例编号	
├─taskId	Long	是		任务ID	
msg	String	是		响应消息	
ts	Long	是		时间戳	1598754678123

请求示例

> 参考：调用方式

快照配额设置任务结果通知

基本信息

URL

> http://host:port/api/v1/instance/dufs-snapshot-quota-set-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用快照配额设置接口后，任务结果会通过该回调接口通知给客户。

请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	Boolean	必须		设置结果标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

仿真授权管理

分页查询营销套件授权码

基本信息

URL

> https://platform.armvm.com/resources/emulation/auth-code-page

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
authCodes	String[]	非必须		授权码集合，集合长度最大为200	["xxx"]
authStatus	Integer	非必须		授权码绑定状态，0-未绑定，1-已绑定	1
instanceCodes	String[]	非必须		绑定的实例编号集合，集合长度最大为200	["VM192168001001"]
packageName	String	非必须		绑定应用包名	com.example
page	Integer	必须		页码	1
rows	Integer	必须		每页条数，最多200	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├──total	Integer	必须		总数	20
├──totalPage	Integer	必须		总页码	2
├──page	Integer	必须		页码	1
├──rows	Integer	必须		每页数量	10
├──pageData	Object[]	必须			
├────────authCode	String	必须		授权码	7290017192180314133
├────────expireTime	Date	必须		授权到期时间（时间戳）	1598754678123
├────────authStatus	Integer	必须		绑定状态，0-未绑定，1-已绑定	1
├────────activateTime	Date	非必须		绑定时间（时间戳）	1598754678123
├────────packageName	String	非必须		绑定应用包名	com.example
├────────deviceIp	String	非必须		绑定实例的所属物理机IP	192.168.1.1
├────────instanceIp	String	非必须		绑定实例IP	192.168.1.1
├────────instanceCode	String	非必须		绑定实例编号	VM192168001001
├────────createTime	Date	必须		授权码创建时间（时间戳）	1598754678123
msg	String	必须		响应消息	
ts	Long	必须		时间戳	1598754678123

请求示例

> 参考：调用方式

响应示例

```
{
"data": {
  "page": 1,
  "rows": 10,
  "totalPage": 103,
  "total": 1021,
  "pageData": [
   {
    "authCode": "xxx",
    "expireTime": 4102415999000,
    "authStatus": 0,
    "activateTime": null,
    "packageName": null,
    "deviceIp": null,
    "instanceIp": null,
    "instanceCode": null,
    "createTime": 1764558830000
   }
  ]
},
"code": 0,
"msg": "OK",
"ts": "1764729944701"
}
```


异常示例

仿真授权码换绑应用

基本信息

URL

> https://platform.armvm.com/resources/emulation/auth-code-rebinding-application

请求格式

> POST

版本号

> 3

使用场景

> 支持已绑定的授权码换绑其他应用，以下情况会导致换绑失败：
>
> 1. 授权码不存在或授权码已过期
> 2. 授权码未绑定应用
> 3. 换绑应用与授权码当前已绑定应用相同
> 4. 授权码对应实例已绑定换绑应用

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
authCodes	String[]	必须		授权码，最多200个	["xxxx"]
packageName	String	必须		换绑应用包名	com.demo

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├── instanceCode	String	必须		实例编号	VM192168043091
├── authCode	String	必须		本次部署使用的授权码	xxx
├── taskId	Long	必须		任务ID	1001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "instanceCode": "VM192168043091",
      "authCode": "xxxx",
      "taskId": 1001
    }
  ],
  "msg": "",
  "ts": 1668572148662
}
```

异常示例

```
{
  "code": 11103062,
  "msg": "实例不存在",
  "ts": "1668586771490"
}
```

实例部署营销套件

基本信息

URL

> https://platform.armvm.com/resources/instance/deploy-marketing-suite

请求格式

> POST

版本号

> 3

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号，最多200个	["VM192168043091"]
authCodes	String[]	非必须		授权码，最多200个，每个实例每个应用绑定一个授权码， 如果实例已绑定该应用，将不消耗授权码，该参数允许为空	["xxxx"]
appPackage	String	必须		应用包名	com.demo
useMerchantAuthCode	Boolean	非必须		是否自动使用商户下的授权码，默认值为false 当传入true时，对于未绑定指定应用的实例，将自动使用商户下的空闲的授权码进行绑定，无需传入authCodes参数	true

> 操作说明：
>
> 1. 为避免历史数据对营销套件效果的影响，强烈建议在部署前先对目标实例进行恢复出厂操作；
> 2. 部署营销套件需配置代理，可通过“配置网络”功能进行配置；
> 3. 如果选择的实例已针对某个应用部署了营销套件时，会重新部署一遍，也会进行新机操作。
> 4. 部署营销套件会重启实例。

> 温馨提示：平台“实例管理-实例列表”中的“授权码绑定状态”为每分钟定时刷新一次，部署完成后，实例的授权码绑定状态最长延迟1分钟更新，但不影响营销套件的使用。

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		错误码	0
data	Object[]	非必须			
├── instanceCode	String	必须		实例编号	VM192168043091
├── authCode	String	非必须		本次部署使用的授权码，如果实例已经绑定该应用，将不消耗授权码，该参数为空	xxx
├── taskId	Long	必须		任务ID	1001
msg	String	必须		错误信息	
ts	Long	必须		时间戳	1601188620078

请求示例

> 参考：调用方式

响应示例

```
{
  "code": 0,
  "data": [
    {
      "instanceCode": "VM192168043091",
      "authCode": "xxxx",
      "taskId": 1001
    }
  ],
  "msg": "",
  "ts": 1668572148662
}
```

异常示例

```
{
  "code":11103062,
  "msg":"实例不存在",
  "ts":"1668586771490"
}
```

脚本自动化

新建自动化任务

基本信息

URL

> https://platform.armvm.com/resources/rpa-task/create

请求格式

> POST

版本号

> 3

使用场景

> 新建自动化任务，使用该功能需要将实例刷入指定的镜像。

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
instanceCodes	String[]	必须		实例编号集合，最多200个	["VM192168001001", "VM192168001002"]
scriptUrl	String	非必须		autojs脚本链接，长度最大255，scriptUrl和scriptContent必填一个	"https://xxx.com/script.js"
scriptContent	String	非必须		autojs脚本内容，长度最大20000，scriptUrl和scriptContent必填一个	"console.log('hello')"

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务id	12259080139067404
├─instanceCode	String	必须		实例编号	"VM192168001001"
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1747035069000

请求示例

```
{
 "instanceCodes": ["VM192168001001", "VM192168001002"],
 "scriptContent": "console.log('hello')"
}
```

响应示例

```
{
"data": [
    {
        "taskId": 12259080139067404,
        "instanceCode": "VM192168001001"
    },
    {
        "taskId": 12259080139067405,
        "instanceCode": "VM192168001002"
    }
],
"code": 0,
"msg": "OK",
"ts": 1747035069000
}
```


异常示例


```
{
"code": 11100003,
"msg": "参数请求错误，脚本url和脚本内容必须且只能填写一个",
"ts": 1747035069000
}

{
"code": 11103062,
"msg": "实例不存在",
"ts": 1747035069000
}
```


查询自动化任务

基本信息

URL

> https://platform.armvm.com/resources/rpa-task/query

请求格式

> POST

版本号

> 3

使用场景

> 查询自动化任务信息

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long[]	必须		任务ID集合，最多200个	[12259080139067404]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object[]	非必须			
├─taskId	Long	必须		任务ID	12259080139067404
├─instanceCode	String	必须		实例编号	"VM192168001001"
├─taskStatus	String	必须		运行状态：pending、running、completed、failed、cancelled、timeout	"completed"
├─startTime	Long	非必须		运行开始时间（时间戳）	1747035069000
├─endTime	Long	非必须		运行结束时间（时间戳）	1747035069000
├─createTime	Long	必须		创建时间（时间戳）	1747035069000
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1747035069000

请求示例

```
{
"taskIds": [12259080139067404, 12259080139067405]
}
```

响应示例

```
{
"data": [
    {
        "taskId": 12259080139067404,
        "instanceCode": "VM192168001001",
        "taskStatus": "completed",
        "startTime": 1747035069000,
        "endTime": 1747035169000,
        "createTime": 1747035069000
    },
    {
        "taskId": 12259080139067405,
        "instanceCode": "VM192168001002",
        "taskStatus": "running",
        "startTime": 1747035069000,
        "endTime": null,
        "createTime": 1747035069000
    }
],
"code": 0,
"msg": "OK",
"ts": 1747035069000
}
```

异常示例


```
{
"code": 11108081,
"msg": "自动化任务记录不存在",
"ts": 1747035069000
}
```


取消自动化任务

基本信息

URL

> https://platform.armvm.com/resources/rpa-task/cancel

请求格式

> POST

版本号

> 3

使用场景

> 取消指定的自动化任务，支持取消pending和running状态的任务

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskIds	Long[]	必须		任务ID集合，最多200个	[12259080139067404]

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须			
├─failTaskIds	Long[]	非必须		取消失败的任务ID列表	[12259080139067404]
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1747035069000

请求示例


```
{
"taskIds": [12259080139067404, 12259080139067405]
}
```

响应示例


```
{
 "data": {
    "failTaskIds": [12259080139067405]
 },
 "code": 0,
 "msg": "OK",
 "ts": 1747035069000
}
```


异常示例


```
{
 "code": 11108081,
 "msg": "自动化任务记录不存在",
 "ts": 1747035069000
}
```


查询自动化任务日志

基本信息

URL

> https://platform.armvm.com/resources/rpa-task/log-page

请求格式

> POST

版本号

> 3

使用场景

> 查询自动化任务的执行日志

请求参数

body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务ID	12259080139067404
logType	String	非必须		日志类型（可选：worker、user）	"worker"
logLevel	String	非必须		日志级别（可选：debug、info、warning、error）	"info"
page	Integer	必须		页码，从1开始	1
rows	Integer	必须		每页条数，最大100	10

返回参数

名称	类型	是否必须	默认值	备注	示例值
code	Integer	必须		响应编码	0
data	Object	非必须		分页信息	
├─total	Integer	必须		总条数	100
├─page	Integer	必须		当前页码	1
├─rows	Integer	必须		每页条数	10
├─totalPage	Object[]	必须		日志列表	
├─ ├─taskId	Long	必须		任务ID	12259080139067404
├─ ├─logType	String	必须		日志类型（worker、user）	"worker"
├─ ├─logLevel	String	必须		日志级别（debug、info、warning、error）	"info"
├─ ├─logMessage	String	必须		日志消息	"Task started successfully"
├─ ├─logTime	Long	必须		日志时间（时间戳）	1747035069000
msg	String	必须		响应消息	"OK"
ts	Long	必须		时间戳	1747035069000

请求示例

```
{
  "taskId": 12259080139067404,
  "logType": "worker",
  "logLevel": "info",
  "page": 1,
  "rows": 10
}
```

响应示例

```
{
"data": {
    "total": 50,
    "page": 1,
    "rows": 10,
    "totalPage": [
        {
            "taskId": 12259080139067404,
            "logType": "worker",
            "logLevel": "info",
            "logMessage": "Task started successfully",
            "logTime": 1747035069000
        },
        {
            "taskId": 12259080139067404,
            "logType": "worker",
            "logLevel": "debug",
            "logMessage": "Loading script...",
            "logTime": 1747035070000
        }
    ]
},
"code": 0,
"msg": "OK",
"ts": 1747035069000
}
```


异常示例


```
{
"code": 11108081,
"msg": "自动化任务记录不存在",
"ts": 1747035069000
}
```


服务回调

回调接口使用示例

> 以实例重启回调为例，其他回调接口使用方法类似。

配置说明

> 需要客户在客户开放平台配置实例重启通知的回调地址，如接口地址为http://host:port/api/v1/instance/reboot-fb，则配置为http://host:port/api/v1/instance/reboot-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	Boolean	必须		上传结果的标识。true：成功，false：失败	true

客户服务端接收请求示例

Java

```
    // 配置好客户自己的desKey，可在开放平台的账号管理菜单下查看和重置
    private String desKey = "d3b5c2b2e480e9e452c8b9f75fbe192b";
    private String appSecret = "5e5b5f13052dfc53159e2069af12e6cf";

    @PostMapping("http:127.0.0.1:8888/api/v1/instance/reboot-fb")
    public Map<String, Object> instanceReboot(String appkey, String nonce, String s, String auth_ver, @RequestBody
    (required = false) String body) throws Exception {
        // 校验签名
        StringBuilder paramsStr = new StringBuilder();
        paramsStr.append("appkey").append(appkey)
                .append("auth_ver").append(auth_ver)
                .append("nonce").append(nonce)
                .append(appSecret);//接口签名密钥（可配置）
        String sign = DigestUtils.md5Hex(org.apache.commons.codec.binary.StringUtils.getBytesUtf8(paramsStr.toString()));
        if (!sign.equals(s)) {
            // 签名校验不合法
        }

        Map<String, Object> bodyMap = new ObjectMapper().readValue(body, Map.class);
        Long bodyNonce = (Long) bodyMap.get("createTime");
        String msg = (String) bodyMap.get("msg");
        byte[] key = desKey.getBytes("UTF-8");
        ByteBuffer byteBuffer = ByteBuffer.allocate(8);
        byteBuffer.order(ByteOrder.BIG_ENDIAN);
        byteBuffer.putLong(bodyNonce);
        byte[] iv = byteBuffer.array();
        byte[] message = Base64.decodeBase64(msg);
        Cipher cipher = Cipher.getInstance("DESede/CBC/PKCS5Padding");
        SecretKeyFactory skf = SecretKeyFactory.getInstance("DESede");
        SecretKey secretKey = skf.generateSecret(new DESedeKeySpec(key));
        IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
        cipher.init(Cipher.DECRYPT_MODE, secretKey, ivParameterSpec);
        byte[] bytes = cipher.doFinal(message);

        // 获取回调的内容
        String requestBody = new String(bytes, "UTF-8");
        logger.info("回调消息, request body : {}", requestBody);

        // 处理客户自己的业务逻辑

        // 返回响应信息
        Map<String, Object> responseHeader = new HashMap<>();
        responseHeader.put("time", System.currentTimeMillis());
        responseHeader.put("version", "2");
        responseHeader.put("status", 200);

        Map<String, Object> response = new HashMap<>();
        response.put("responseHeader", responseHeader);
        return response;
    }
```

测试请求示例

Java

```
    // PaaS 平台分配给第三方用户的接口Key
    String appkey = "50b13132bb394901f151bc45";
    // 接口版本号，现在有2和3两种版本的接口
    String auth_ver = "2";
    // 当前的时间戳
    long nonce = new Date().getTime();
```

```
    // 方法签名，生成示例见'方法签名生成示例'
    String s = "efabcaca7c66d629a639cee37d924813";
    //desKey 3DES秘钥（可配置）,根据商户平台实际生成的替换此值
    String desKey = "d3b5c2b2e480e9e452c8b9f75fb1qazs";
    // 拼接公共参数到URI
    String apiUrl = "http://host:port/api/v1/instance/reboot-fb?s=" + s + "&appkey=" + appkey + "&auth_ver=" + auth_ver
+ "&nonce=" + nonce;

    // 填装请求参数
    Map param = new HashMap();
    param.put("taskId", 1);
    param.put("result", true);
    param.put("msg", "ok");
    param.put("padCode", "VM123");

    // 请求参数转化为JSON字符串
    String requestParam = new Gson().toJson(param);

    // 请求参数加密
    Cipher cipher = Cipher.getInstance("DESede/CBC/PKCS5Padding");
    SecretKeyFactory skf = SecretKeyFactory.getInstance("DESede");
    SecretKey secretKey = skf.generateSecret(new DESedeKeySpec(desKey.getBytes("utf-8")));
    ByteBuffer byteBuffer = ByteBuffer.allocate(8);
    byteBuffer.order(ByteOrder.BIG_ENDIAN);
    byteBuffer.putLong(nonce);
    byte[] iv = byteBuffer.array();
    IvParameterSpec ivParameterSpec = new IvParameterSpec(iv);
    cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivParameterSpec);
    byte[] cipherText = cipher.doFinal(requestParam.getBytes("utf-8"));
    String msg = Base64.encodeBase64String(cipherText);
    Map map = new HashMap();
    map.put("createTime", nonce);
    map.put("msg", msg);
    // 转化为JSON字符串
    String encryptBody = new Gson().toJson(map);

    // 构造请求
    URL url = new URL(apiUrl);
    HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
    // 连接超时时间
    httpURLConnection.setConnectTimeout(1000);
    // 读取结果超时时间
    httpURLConnection.setReadTimeout(1000);
    httpURLConnection.setDoInput(true);
    httpURLConnection.setDoOutput(true);
    httpURLConnection.setUseCaches(false);
    httpURLConnection.setRequestProperty("Content-type", "application/json;charset=utf-8");
    httpURLConnection.setRequestMethod("POST");
    if ("https".equalsIgnoreCase(url.getProtocol())) {
        HttpsURLConnection husn = (HttpsURLConnection) httpURLConnection;
        husn.setSSLSocketFactory(new BaseHttpSSLSocketFactory());
        //解决由于服务器证书问题导致HTTPS无法访问的情况
        husn.setHostnameVerifier(new BaseHttpSSLSocketFactory.TrustAnyHostnameVerifier());
        httpURLConnection = husn;
    }
    // 得到请求的输出流对象
    OutputStreamWriter out = new OutputStreamWriter(httpURLConnection.getOutputStream());
    // 把数据写入请求的Body
    out.write(encryptBody);
    out.flush();
    out.close();
```

```
    String result = "";
    try (InputStream in = 200 == httpURLConnection.getResponseCode() ? httpURLConnection.getInputStream() :
httpURLConnection.getErrorStream();) {
        byte[] buf = new byte[1024];
        int length = 0;
        ByteArrayOutputStream bout = new ByteArrayOutputStream();
        while ((length = in.read(buf, 0, buf.length)) > 0) {
            bout.write(buf, 0, length);
        }
        bout.flush();
        result = new String(bout.toByteArray(), StandardCharsets.UTF_8);
    } catch (Exception e) {
        throw e;
    } finally {
        if (null != httpURLConnection) {
            httpURLConnection.disconnect();
        }
    }
    int responseCode = httpURLConnection.getResponseCode();
    if (responseCode == 200) {
        System.out.println("调用成功,接口返回结果:\n" + result);
    } else {
        System.out.println("调用失败");
    }
```

请求参数示例

```
{
  "taskId": 1,
  "padCode": "VM123",
  "result": true,
  "msg": "ok"
  }
```

应用安装通知

基本信息

URL

> http://host:port/api/v1/app/install-fb

请求格式

> POST

版本号

> 3

使用场景

> 应用安装任务的结果会通过该回调接口通知。

配置说明

> 需要在开放平台配置应用安装通知的回调地址，如接口地址为http://host:port/api/v1/app/install-fb，则配置为http://host:port/api/v1/app/install-fb

请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
apps	Object	必须		应用信息	
├─appId	Long	必须		应用ID	10001
├─padCode	String	必须		实例编码	VM123
├─appName	String	必须		应用名称	demo
├─pkgName	String	必须		应用包名	com.demo
├─result	boolean	必须		任务结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

应用卸载通知

基本信息

URL

> http://host:port/api/v1/app/uninstall-fb

请求格式

> POST

版本号

> 3

使用场景

> 应用卸载任务的结果会通过该回调接口通知。

配置说明

> 需要在开放平台配置应用卸载通知的回调地址，如接口地址为http://host:port/api/v1/app/uninstall-fb，则配置为http://host:port/api/v1/app/uninstall-fb

请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
apps	Object	必须		应用信息	
├──appId	Long	必须		应用ID	10001
├──padCode	String	必须		实例编码	VM123
├──appName	String	必须		应用名称	demo
├──pkgName	String	必须		应用包名	com.demo
├──result	boolean	必须		任务结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

保存应用通知

基本信息

URL

> http://host:port/api/v1/app/save-fb

请求格式

> POST

版本号

> 3

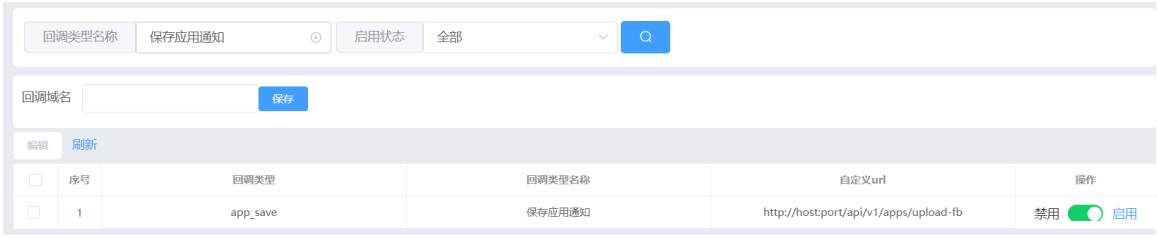
使用场景

> 调用百度云游戏应用上传接口，应用上传情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置保存应用通知的回调地址，如接口地址为http://host:port/api/v1/app/save-fb，则配置为http://host:port/api/v1/app/save-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	非必须		任务编号，只有在应用上传接口中传入taskId时有值	1
apps	Object	必须		应用信息	
├─appId	Long	必须		应用ID	10001
├─appName	String	必须		应用名称	demo
├─pkgName	String	必须		包名	com.demo
├─result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

应用启动通知

基本信息

URL

> http://host:port/api/v1/app/start-fb

请求格式

> POST

版本号

> 3

使用场景

> 当用户调用应用启停接口后，应用启动情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置应用启动通知的回调地址，如接口地址为http://host:port/api/v1/app/start-fb，则配置为http://host:port/api/v1/app/start-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
packageName	String	必须		应用包名	com.demo
operateType	String	必须		startApp：应用启动	startApp
taskStatus	String	必须		任务结果：7： 成功 ； 8： 失败	7
apps	Object	必须		应用信息	
├─appId	Long	必须		应用ID	10001
├─padCode	String	必须		实例编码	VM123
├─appName	String	必须		应用名称	demo
├─pkgName	String	必须		包名	com.demo
├─result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

应用停止通知

基本信息

URL

> http://host:port/api/v1/app/stop-fb

请求格式

> POST

版本号

> 3

使用场景

> 当用户调用应用启停接口后，应用停止情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置应用停止通知的回调地址，如接口地址为http://host:port/api/v1/app/stop-fb，则配置为http://host:port/api/v1/app/stop-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
packageName	String	必须		应用包名	com.demo
operateType	String	必须		killApp：应用停止	killApp
taskStatus	String	必须		任务结果：7： 成功 ； 8： 失败	7
apps	Object	必须		应用信息	
├─appId	Long	必须		应用ID	10001
├─padCode	Long	必须		实例编码	VM123
├─appName	String	必须		应用名称	demo
├─pkgName	String	必须		包名	com.demo
├─result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例解绑通知

基本信息

URL

> http://host:port/api/v1/instance/unbind-fb

请求格式

> POST

版本号

> 3

使用场景

> 客户调用百度云游戏实例解绑接口后，实例解绑情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例解绑通知的回调地址，如接口地址为http://host:port/api/v1/instance/unbind-fb，则配置为http://host:port/api/v1/instance/unbind-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
uuid	String	非必须		uuid，业务类型非 OEM 时返回	123456
uid	String	非必须		用户名，业务类型为 OEM 时返回	userName
padCode	String	必须		实例编码	VM123123123123
instanceCode	String	必须		实例编码	VM123123123123
sessionId	String	必须		会话ID	JKWEUNBL6RX3WCBUQ92AGYCU9MLX0WTB
packageName	String	非必须		应用包名	com.example.demo
endTimestamp	Long	必须		完成解绑操作的时间戳	1669020092256
errorCode	String	非必须		解绑结果错误码	0
errorInfo	String	非必须		解绑结果错误信息	success
businessType	String	非必须		业务类型：oem、cloud_phone、cloud_app 等	cloud_phone

使用示例

> 参考：回调接口使用示例

实例重置通知

基本信息

URL

> http://host:port/api/v1/instance/reset-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用百度云游戏实例重置接口后，实例重置情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例重置通知的回调地址，如接口地址为http://host:port/api/v1/instance/reset-fb，则配置为http://host:port/api/v1/instance/reset-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例重启通知

基本信息

URL

> http://host:port/api/v1/instance/reboot-fb

请求格式

> POST

版本号

> 3

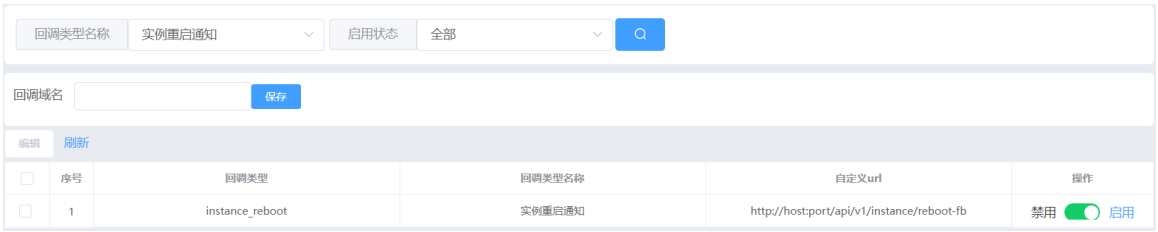
使用场景

> 调用百度云游戏实例重启接口后，实例重启情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例重启通知的回调地址，如接口地址为http://host:port/api/v1/instance/reboot-fb，则配置为http://host:port/api/v1/instance/reboot-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例执行脚本通知

基本信息

URL

> http://host:port/api/v1/instance/execute-script-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用百度云游戏实例执行脚本接口后，脚本执行情况会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例执行脚本通知的回调地址，如接口地址为http://host:port/api/v1/instance/execute-script-fb，则配置为http://host:port/api/v1/instance/execute-script-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	Boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

上传文件到实例通知

基本信息

URL

> http://host:port/api/v1/instance/file-upload-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用百度云游戏上传文件到实例接口后，文件上传结果会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置上传文件到实例通知的回调地址，如接口地址为http://host:port/api/v1/instance/file-upload-fb，则配置为http://host:port/api/v1/instance/file-upload-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	Boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

桌面推荐图标刷新通知

基本信息

URL

> http://host:port/api/v1/instance/recommend-app-icon-refresh

请求格式

> POST

版本号

> 3

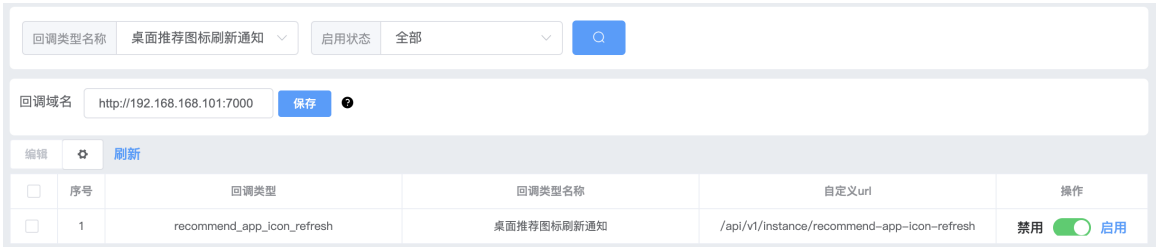
使用场景

> 调用百度云桌面推荐图标刷新，刷新结果会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置桌面推荐图标刷新通知的回调地址，如接口地址为http://host:port/api/v1/instance/recommend-app-icon-refresh，则配置为http://host:port/api/v1/instance/recommend-app-icon-refresh

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM192168043080
msg	String	必须		返回信息	ok
result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例备份通知

基本信息

URL

> http://host:port/api/v1/instance/backup-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用百度云实例备份接口后，实例备份的结果会通过该回调接口通知。

配置说明

> 需要在开放平台配置实例备份通知的回调地址，如接口地址为http://host:port/api/v1/instance/backup-fb，则配置为http://host:port/api/v1/instance/backup-fb

请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	boolean	必须		任务结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例还原通知

基本信息

URL

> http://host:port/api/v1/instance/restore-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用百度云游戏实例还原接口后，执行结果会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例还原通知的回调地址，如接口地址为http://host:port/api/v1/instance/restore-fb，则配置为http://host:port/api/v1/instance/restore-fb

请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例分辨率调整通知

基本信息

URL

> http://host:port/api/v1/instance/resolution-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用实例分辨率调整接口后，调整结果会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例分辨率调整通知的回调地址，如接口地址为http://host:port/api/v1/instance/resolution-fb，则配置为http://host:port/api/v1/instance/resolution-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	Boolean	必须		执行结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

进程死亡告警通知

基本信息

URL

> http://host:port/api/v1/monitor/process-died-alarm-fb

请求格式

> POST

版本号

> 3

使用场景

> 实例应用进程异常死亡后，进程死亡信息会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置进程死亡告警通知的回调地址，如接口地址为http://host:port/api/v1/monitor/process-died-alarm-fb，则配置为http://host:port/api/v1/monitor/process-died-alarm-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
instanceCode	String	必须		实例编码	VM123
content	Object	必须			
├──packageName	String	必须		应用包名	com.example.app
├──reason	String	必须		进程死亡原因；参考reason枚举表；	process other exception died
├──controlStatus	String	必须		实例受控状态；control：受控，online：非受控；	online
├──uuid	String	必须		实例绑定的uuid	
├──recordTime	String	必须		进程死亡时间	2024-10-30 12:00:00

reason枚举

reason	含义
anr	进程anr被终止
bg anr	后台进程anr被终止
crash	进程崩溃被终止
start timeout	进程启动超时被终止
depends on provider	所绑定的provider已销毁所以被杀
lmk-kill	进程被Low Memory Killer终止

使用示例

> 参考：回调接口使用示例

实例文件下载通知

基本信息

URL

> http://host:port/api/v1/instance/file-download-fb

请求格式

> POST

版本号

> 3

使用场景

> 请求实例文件下载后，任务将进行异步执行，执行完毕后会通过该回调接口通知到客户；

配置说明

> 需要客户在客户开放平台配置实例文件下载通知的回调地址，如接口地址为http://host:port/api/v1/instance/file-download-fb，则配置为http://host:port/api/v1/instance/file-download-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1000
instanceCode	String	必须		实例编码	"VM192168001001"
result	Boolean	必须		任务执行结果；true：成功，false：失败	true
msg	String	必须		返回信息	"https://callback.example.com/file.txt"

使用示例

> 参考：回调接口使用示例

实例状态变更通知

基本信息

URL

> http://host:port/api/v1/instance/state-change-fb

请求格式

> POST

版本号

> 3

使用场景

> 虚机状态变更，会将变更的状态回调接口通知到客户，变更状态包括在线状态、受控状态、维护状态以及PaaS维护状态

配置说明

> 需要客户在客户开放平台配置实例状态变更通知的回调地址，如接口地址为http://host:port/api/v1/instance/state-change-fb，则配置为http://host:port/api/v1/instance/state-change-fb

配置示例



请求参数

Body

请求参数以 JSON 数组的形式传递，每个数组元素是一个对象，包含以下字段：

名称	类型	是否必须	默认值	备注	示例值
instanceCode	String	必须		实例编码	"VM192168001001"
stateType	String	必须		实例状态类型： instanceStatus-在线状态； controlStatus-受控状态； maintainStatus-维护状态； paasMaintainStatus-PaaS维护状态； malfunctionStatus-故障状态；	instanceStatus
stateValue	String	必须		实例状态值： instanceStatus：0-离线，1-在线； controlStatus：1-在线，2-控制； maintainStatus：0-正常，1-维护； paasMaintainStatus：0-正常，1-维护； malfunctionStatus：0-正常，1-故障；	0

说明：

- PaaS维护状态为上游服务进行操作时，对实例进行维护的状态，避免操作对客户的实例以及业务造成影响；
- 故障状态定义：实例离线超过2分钟。

示例：

```
[
  {
    "instanceCode": "VM192168001001",
    "stateType": "instanceStatus",
    "stateValue": "1"
  },
  {
    "instanceCode": "VM192168001002",
    "stateType": "instanceStatus",
    "stateValue": "1"
  }
]
```

使用示例

> 参考：回调接口使用示例

实例会话受控状态变更通知

基本信息

URL

> http://host:port/api/v1/instance/control-status-change-fb

请求格式

> POST

版本号

> 3

使用场景

> 当实例会话受控状态变更时，将会通过该回调接口通知到客户；

配置说明

> 需要客户在客户开放平台配置受控状态变更通知的回调地址，如接口地址为http://host:port/api/v1/instance/control-status-change-fb，则配置为http://host:port/api/v1/instance/control-status-change-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
changeInfos	Object[]	必须			
├─uuid	String	必须		用户标识	
├─instanceCode	String	必须		实例编号	"VM192168001001"
├─serverToken	String	必须		会话的serverToken	
├─controlStatus	String	必须		受控状态；online-非受控，control-受控	"control"
changeTime	Long	必须		变更时间	1740573508898

使用示例

> 参考：回调接口使用示例

实例对拷结果通知

基本信息

URL

> http://host:port/api/v1/instance/data-copy-fb

请求格式

> POST

版本号

> 3

使用场景

> 实例对拷任务完成后，对拷结果会通过该接口回调给客户。

配置说明

> 需要客户在客户开放平台配置实例对拷结果通知的回调地址，如接口地址为http://host:port/api/v1/instance/data-copy-fb，则配置为http://host:port/api/v1/instance/data-copy-fb

配置示例



请求参数

Body

名称	类型	是否必须	备注	示例值
taskId	Long	必须	任务ID	1001
sourceInstanceCode	String	必须	源实例编号	VM192168001001
targetInstanceCode	String	必须	目标实例编号	VM192168001002
result	Boolean	必须	任务结果的标识。true：成功，false：失败	true
msg	String	非必须	附带信息，一般是失败信息	

使用示例

> 参考：回调接口使用示例

消息栏消息通知

基本信息

URL

> http://host:port/api/v1/instance/message-bar-fb

请求格式

> POST

版本号

> 3

使用场景

> 云手机内应用接收到消息后，消息内容通过该接口回调给客户

配置说明

> 需要客户在客户开放平台配置消息栏消息通知的回调地址，如接口地址为http://host:port/api/v1/instance/message-bar-fb，则配置为http://host:port/api/v1/instance/message-bar-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例
body	Object[]	必须			
├─instanceCode	String	必须		实例编码	VM123
├─packageName	String	必须		应用包名	
├─createTime	Date	必须		消息创建时间	
├─title	String	必须		标题	
├─content	String	必须		内容	

使用示例

> 参考：回调接口使用示例

物理机GPU告警通知

基本信息

URL

> http://host:port/api/v1/device/gpu-alarm-fb

请求格式

> POST

版本号

> 3

使用场景

> 物理机GPU告警，会通过该回调接口通知。

配置说明

> 需要在开放平台配置物理机GPU告警通知的回调地址，如接口地址为http://host:port/api/v1/device/gpu-alarm-fb，则配置为http://host:port/api/v1/device/gpu-alarm-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
recordTime	Long	必须		发生事件（时间戳）	1663664302000
gpuUsage	String	必须		GPU使用量	
instances	Object[]	必须		实例信息	
├── uuid	String	必须			
├── instanceCode	String	必须		实例编号	VM123
├── appPackage	String	必须		应用包名	com.demo
├── controlStatus	String	必须		控制状态：online-未受控，control受控	online

使用示例

> 参考：回调接口使用示例

实例分配通知

基本信息

URL

> http://host:port/api/v1/instance/allocate-fb

请求格式

> POST

版本号

> 3

使用场景

> 实例分配结果，会通过该回调接口通知。

配置说明

> 需要在开放平台配置实例分配通知的回调地址，如接口地址为http://host:port/api/v1/instance/allocate-fb，则配置为http://host:port/api/v1/instance/allocate-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
allocateTime	Date	必须		分配时间	2025-09-17 00:00:00
recycleTime	Date	必须		回收时间	2025-09-17 00:00:00
instanceCodes	String[]	必须		实例列表	["VM123"]

使用示例

> 参考：回调接口使用示例

实例一键新机通知

基本信息

URL

> http://host:port/api/v1/instance/new-model-fb

请求格式

> POST

版本号

> 3

使用场景

> 实例一键新机结果，会通过该回调接口通知。

配置说明

> 需要在开放平台配置实例一键新机通知的回调地址，如接口地址为http://host:port/api/v1/instance/new-model-fb，则配置为http://host:port/api/v1/instance/new-model-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
msg	String	必须		返回信息	ok
result	boolean	必须		结果，true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

实例池数量变更通知

基本信息

URL

> http://host:port/api/v1/instance-pool/change

请求格式

> POST

版本号

> 3

使用场景

> 实例池数量变更，会通过该回调接口通知。

配置说明

> 需要在开放平台配置实例分配通知的回调地址，如接口地址为http://host:port/api/v1/instance-pool/change，则配置为http://host:port/api/v1/instance-pool/change

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
merchantPoolNo	Long	必须		涉及实例池编号	1001
changeType	String	必须		变更类型 - 实例池间调度 - 纯新增 - 纯减少	"纯新增"
changeStateValue	String	必须		变更状态值 - 1-新增 - 0-减少	"1"
changeCount	Integer	必须		变更数量	5
instances	Instance[]	必须		实例列表	
├── originMerchantPoolNo	Long	否		当实例值数量新增时有值，否则为空	1000
├── targetMerchantPoolNo	Long	否		当实例值数量减少时有值，否则为空	1002
├── instanceCode	String	必须		实例编号	"VM001"
├── deviceIp	String	必须		物理机IP	"192.168.1.100"
├── specification	String	必须		实例规格	

使用示例

> 参考：回调接口使用示例

实例回收通知

基本信息

URL

> http://host:port/api/v1/instance/retract-fb

请求格式

> POST

版本号

> 3

使用场景

> 实例回收结果，会通过该回调接口通知。

配置说明

> 需要在开放平台配置实例回收通知的回调地址，如接口地址为http://host:port/api/v1/instance/retract-fb，则配置为http://host:port/api/v1/instance/retract-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
recycleTime	Date	必须		回收时间	2025-09-17 00:00:00
instanceCodes	String[]	必须		实例编码列表	["VM123"]

使用示例

> 参考：回调接口使用示例

实例ROOT/UNROOT结果通知

基本信息

URL

> http://host:port/api/v1/instance/set-root-fb

请求格式

> POST

版本号

> 3

使用场景

> 实例ROOT/UNROOT结果，会通过该回调接口通知。

配置说明

> 需要在开放平台配置实例分配通知的回调地址，如接口地址为http://host:port/api/v1/instance/set-root-fb，则配置为http://host:port/api/v1/instance/set-root-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务 ID	123456
instanceCode	String	必须		实例编号	"VM001"
operationType	String	必须		操作方式 - root: root操作 - unroot: unroot操作	"root"
taskStatus	String	必须		任务状态 - wait_execute: 待执行 - executing: 执行中 - success: 执行成功 - fail: 执行失败	"success"
taskResult	String	必须		任务结果信息	

使用示例

> 参考：回调接口使用示例

资源告警通知

基本信息

URL

> http://host:port/api/v1/resource/resource-alarm-fb

请求格式

> POST

版本号

> 3

使用场景

> 资源告警，会通过该回调接口通知。

配置说明

> 需要在开放平台配置资源告警通知的回调地址，如接口地址为http://host:port/api/v1/resource/resource-alarm-fb，则配置为http://host:port/api/v1/resource/resource-alarm-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
alarmStrategyId	Long	必须		告警策略Id	
alarmStrategyName	String	必须		告警策略名称	
idcCode	String	必须		机房编号	
idcName	String	必须		机房名称	
alarmResourceType	String	必须		资源告警类型:OSS(对象存储)/DuFS(DuFS存储)/Bandwidth(带宽用量)	
alarmMetrics	String	必须		告警维度	
alarmThreshold	Integer	必须		告警阈值	
usage	Long	必须		用量	
total	Long	必须		总量	
alarmTime	Long	必须		告警时间	
subsidiaryAttributes	Object	必须		附属属性	
├── bucketName	String	非必须		对象存储桶名（仅OSS存储需要）	

使用示例

> 参考：回调接口使用示例

应用更新通知

基本信息

URL

> https://host:port/api/v1/app/update-notify-fb

请求格式

> POST

版本号

> 3

使用场景

> 应用更新检测时发现检测中的应用存在更新时发送该通知

配置说明

> 需要客户在客户开放平台配置保存应用更新通知的回调地址，如接口地址为http://host:port/api/v1/app/update-notify-fb，则配置为http://host:port/api/v1/app/update-notify-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
appName	String	必须		应用名称	原神
appPackage	String	必须		应用包名	com.miHoYo.Yuanshen
appVersion	String	必须		应用版本	3.0.0

使用示例

> 参考：回调接口使用示例

实例部署营销套件通知

基本信息

URL

> http://host:port/api/v1/instance/deploy-marketing-suite-fb

请求格式

> POST

版本号

> 3

使用场景

> 调用实例部署营销套件接口后，任务执行结果会通过该回调接口通知给客户。

配置说明

> 需要客户在客户开放平台配置实例部署营销套件通知的回调地址，如接口地址为http://host:port/api/v1/instance/deploy-marketing-suite-fb，则配置为http://host:port/api/v1/instance/deploy-marketing-suite-fb

配置示例



请求参数

Body

名称	类型	是否必须	默认值	备注	示例值
taskId	Long	必须		任务编号	1
padCode	String	必须		实例编码	VM123
authCode	String	非必须		本次部署使用的授权码，如果实例已经绑定该应用，部署过程中不消耗授权码，该参数为空	xxx
appPackage	String	必须		本次部署的应用	com.example
msg	String	必须		返回信息	ok
result	Boolean	必须		上传结果的标识。true：成功，false：失败	true

使用示例

> 参考：回调接口使用示例

物理机状态变更通知

基本信息

URL

> http://host:port/api/v1/device/state-change-fb

请求格式

> POST

版本号

> 3

使用场景

> 物理机状态变更后，会通过该回调接口通知给客户，变更状态包括在线状态、维护状态

配置说明

> 需要客户在客户开放平台配置物理状态变更通知的回调地址，如接口地址为http://host:port/api/v1/device/state-change-fb，则配置为http://host:port/api/v1/device/state-change-fb

配置示例



请求参数

Body

请求参数以 JSON 数组的形式传递，每个数组元素是一个对象，包含以下字段： | 名称 | 类型 | 是否必须 | 默认值 | 备注 | 示例值 | | ---------- | ------ | -------- | ------ | ---------- | | deviceIp | String | 必须 | | 物理机IP | "192.0.0.1" | | stateType | String | 必须 | | 实例状态类型：
deviceStatus-在线状态；
maintainStatus-维护状态； | deviceStatus | | stateValue | String | 必须 | | 实例状态值：
deviceStatus：0-离线，1-在线；
maintainStatus：0-正常，1-维护； | 0 |

示例：

```
[
  {
    "deviceIp": "192.0.0.1",
    "stateType": "deviceStatus",
    "stateValue": "1"
  },
  {
    "deviceIp": "192.0.0.2",
    "stateType": "deviceStatus",
    "stateValue": "0"
  }
]
```

使用示例

> 参考：回调接口使用示例

返回结果

版本一

返回参数

名称	类型	是否必须	默认值	备注	示例值
resultCode	Integer	必须		操作结果代码，0代表成功 -1代表失败 9999代表系统错误	0
resultInfo	String	必须		操作具体描述以及数据封装	

版本二

返回参数

名称	类型	是否必须	默认值	备注	示例值	
code	Integer	必须		响应编码	0	
data	Object[]	非必须				
msg	String	非必须		响应消息		
ts	Long	必须		十三位时间戳	1598754678123	

错误码

错误码	错误信息	备注
11100050	系统异常，请联系客服人员	
11101100	商户信息异常	
11101118	商户类型错误,只允许云手机客户使用	
11102027	serverToken无效	
11102032	serverToken无效，[serverToken]	
11103062	实例不存在	
11103078	实例不属于此用户	
11103103	无设备可绑定	
11103180	实例池不存在	
11103183	实例池中无实例	
11103184	实例编号和实例池编码不能同时为空	
11103185	实例池不存在或无实例	
11103190	实例池不存在，[实例池编号]	
11108003	工作任务不存在	
11108009	找不到脚本[脚本名称],请确定是否有配置	
11109002	内部服务通信失败	

云手机Web SDK

百度云手机Web SDK接入说明

1 SDK概述

主要功能是为Web赋予直连云手机的能力，用户在管理控制台配置好云手机之后，Web端通过SDK进行直连操作。

1.1 运行环境

支持现代浏览器PC、H5端，以及小程序，公众号等。

1.2 环境要求

1、IOS 12.0及以上的系统版本
2、Android浏览器：需要支持webRTC的浏览器内核
3、PC Web浏览器

- 运行js模式需要内核为Chrome58+的浏览器
- 运行webrtc模式需要Chrome 71+的浏览器
- 运行js模式的webcodecs解码需要Chrome 94+、ios 16.4+的浏览器
- 运行js模式的webcodecs的h265解码需要Chrome 105+、ios 16.4+的浏览器

2 更新历史

版本号	发布时间	发布内容	sdk下载
1.43.0	2025-04-08	去除网关模式 优化webcodces解码性能	点击下载
1.42.0	2025-01-02	增加三线切换功能 增加摄像头推送视频流的方式	点击下载
1.41.0	2024-12-25	群控模式增加对接apaas服务	点击下载
1.40.0	2024-12-09	优化埋点、优化滚动功能、优化按键功能	点击下载
1.39.0	2024-11-25	增加推送摄像头流的接口	点击下载
1.38.0	2024-11-07	增加wasm解码、webcodecs适配ios云	点击下载
1.37.0	2024-09-30	优化移动端横屏展示效果	点击下载
1.36.0	2024-08-28	摄像头支持画质调整、优化埋点上报	点击下载
1.35.0	2024-08-14	webcodecs支持h265解码、优化图片上传功能、解决一些已知问题	点击下载
1.34.0	2024-06-21	优化移动端本地输入法问题	点击下载
1.33.0	2024-05-21	自适应屏幕设置为可配置、支持摇一摇功能	点击下载
1.32.0	2024-04-29	优化网络切换的处理、优化摄像头切换处理	点击下载
1.31.0	2024-04-25	应用默认键位支持自定义WASD按键处理	点击下载

1.30.0	2024-04-02	支持摄像头/麦克风的切换与启停	点击下载
1.29.0	2024-03-20	解决一些已知问题、增加离开或切换tab时浏览器是否静音开关机制、js解码库升级到3.0版本	点击下载
1.28.0	2024-01-24	优化横竖屏逻辑、使用webworker执行心跳	点击下载
1.27.1	2024-01-09	优化webcodecs关于软硬编码的处理逻辑	点击下载
1.27.0	2024-01-04	统一消息透传编码的方式、优化埋点等	点击下载
1.26.0	2023-12-20	解决一些已知的问题	点击下载
1.25.0	2023-11-16	sdk传信息给云机粘贴板，增加是否顺便粘贴到输入框中	
1.24.0	2023-11-3	支持切换本地或者云端输入法	
1.23.0	2023-10-19	支持滚轮操作、js解码库升级为2.0版本	
1.22.0	2023-10-07	增加js模式的webcodecs硬解码	
1.21.0	2023-8-29	支持三键导航设置（**需要配套remote-play：4.22.0，vmbroker 2.9.0，镜像：2.25.0**）	
1.20.0	2023-8-21	优化wasd按键不准以及延迟等问题	
1.19.0	2023-7-21	优化键盘映射问题	
1.18.0	2023-7-06	支持软硬键盘切换 （**需要配合 remote-play 4.18.0，vmbroker 2.7.0 使用**）	
1.17.0	2023-6-16	新增接口返回当前实例ID	
1.14.0	2023-5-11	优化运行信息回调	
1.13.0	2023-5-08	优化ios12不能投屏的问题	
1.12.0	2023-4-20	适配电视盒子	
1.11.0	2023-3-23	优化重连功能	
1.10.0	2023-3-23	支持消息透传、错误上报	
1.9.0	2023-2-24	支持群控、多路推流、关闭声音	
1.8.0	2023-2-08	支持键盘设置	
1.7.10	2023-2-03	设备申请接口增加支持模板机调用	
1.7.9	2022-12-15	webrtc p2p模式静音回调优化	
1.7.8	2022-12-06	优化自动切换推流模式功能	
1.7.7	2022-11-17	部分回调函数优化	
1.7.6	2022-11-03	支持云手机APP通话	
1.7.5	2022-10-	支持云手机场景	

	20		
1.7.4	2022-09-30	重连逻辑优化	
1.7.3	2022-09-19	支持云端粘贴板回调	
1.7.2	2022-09-01	支持断点续连功能	
1.7.1	2022-08-17	摄像头和其他功能优化	
1.7.0	2022-08-03	支持接入aPaas	

3 接入准备

1.登录智能云官网账号并完成企业认证，进入云手机管理控制台-概览页面，开通云手机服务。

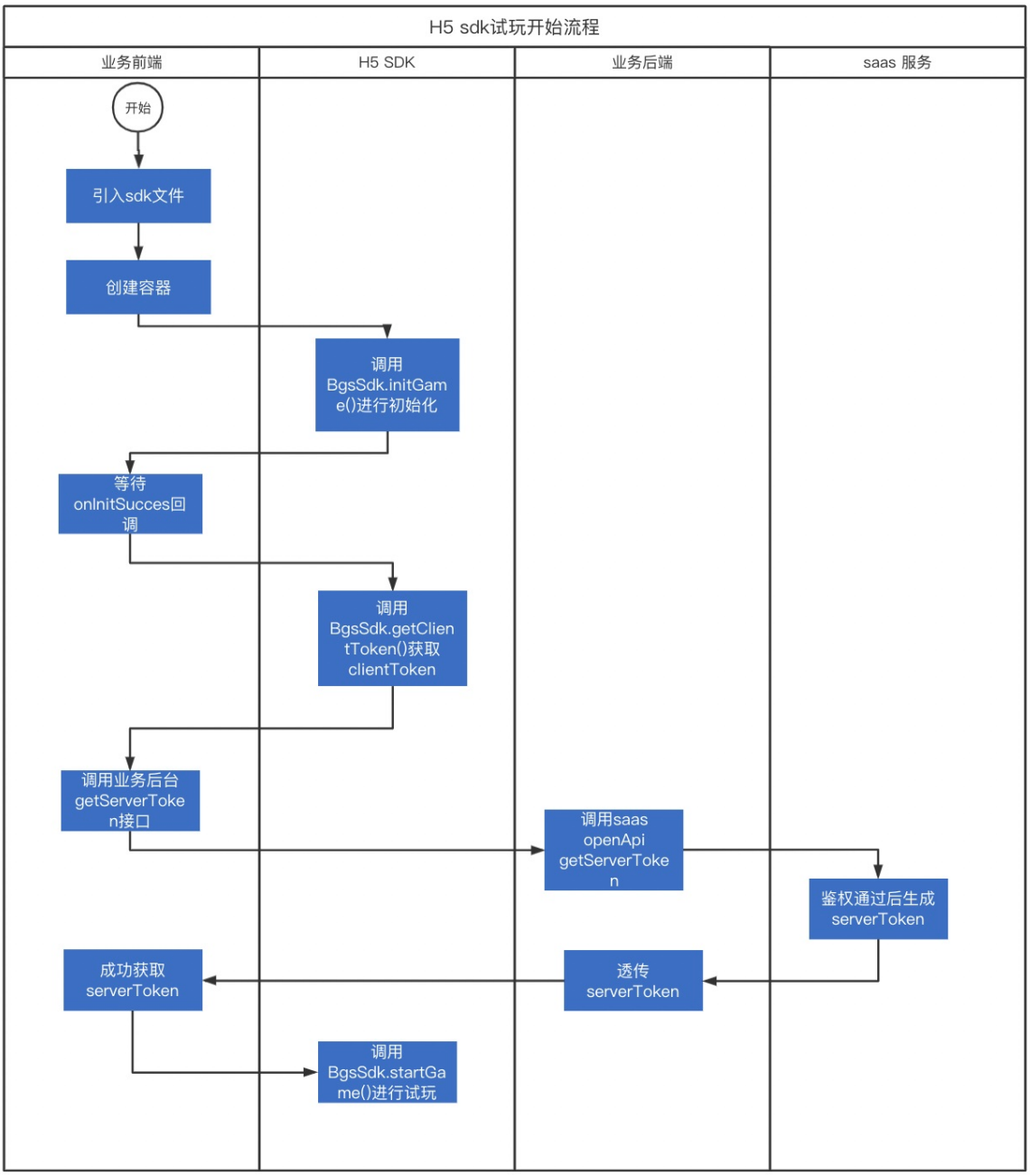
2.购买云手机、或申请试用云手机

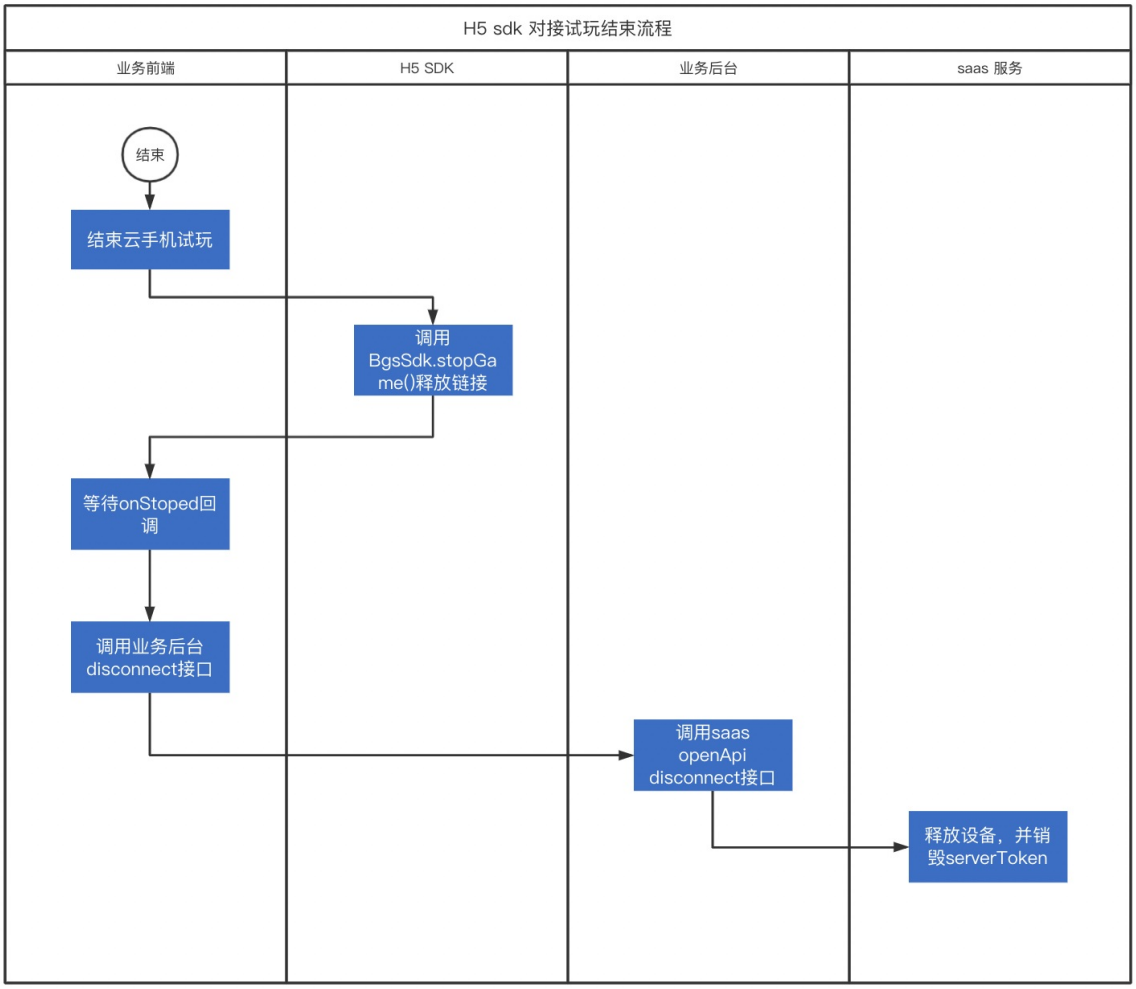
3.进入云手机资源所在的项目，点击“更多管理功能”跳转至项目管理平台

4.跳转平台后，通过账号基础信息获取appKey, appSecret, desKey。

4 接入流程

对接流程：


H5 sdk试玩开始流程
业务前端
H5 SDK
业务后端
saas 服务
开始
引入sdk文件
创建容器
调用BgsSdk.initGame()进行初始化
等待onInitSucces回调
调用BgsSdk.getClientToken()获取clientToken
调用业务后台getServerToken接口
调用saas openApi getServerToken
鉴权通过后生成serverToken
成功获取serverToken
透传serverToken
调用BgsSdk.startGame()进行试玩



特别提醒：建议调用paas api放在自己的服务端实现，避免在客户端泄露appKey, appSecret, desKey等信息。

> 备注：
> 业务前端：客户实现的业务前端系统。
> H5 SDK：百度云手机提供的云手机在h5 页面上使用的对接 sdk。
> 业务后台：客户实现的业务后台系统。
> saas 服务：百度云手机提供的云手机saas管理平台管理服务，接口能力。

4.1 开始试玩流程说明

1、H5 sdk文件请联系云手机对接人员获取最新版本。

2、在页面上创建一个容器

```
<div class="phoneBox" id="phoneBox" style="background-color: grey;width: 100%;height: 100%">
```

3、调用BgsSdk.initPhone()接口进行初始化。

4、调用业务getServerToken接口，传递uuid为参数.

5、业务后台收到getServerToken的请求后，使用saas平台提供的appkey appsecret deskey 进行签名和参数加密，然后调用saas的openApi getServerToken 来获取到serverToken，并将serverToken透传给业务前端。

6、业务前端得到serverToken后，调用BgsSdk.startPhone()进行开始云手机试玩。

4.2 结束试玩流程说明

1、业务前端调用BgsSdk.stopPhone()接口，结束本次云手机链接。

2、断开和云手机的链接后会触发onStoped回调，

4.3 示例代码

```
<!DOCTYPE html>
<html>

<head>
    <title>webrtc</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <meta name="format-detection" content="telephone=no, email=no" />
    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
    <meta name="apple-mobile-web-app-capable" content="yes" />
    <!-- 删除苹果默认的工具栏和菜单栏 -->
    <meta name="apple-mobile-web-app-status-bar-style" content="black" />
    <!-- 设置苹果工具栏颜-->
    <meta name="format-detection" content="telphone=no, email=no" />
    <!-- 忽略页面中的数字识别为电话，忽略email识别 -->
    <!-- 启用360浏览器的极速模(webkit) -->
    <meta name="renderer" content="webkit">
    <!-- 避免IE使用兼容模式 -->
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <!-- 针对手持设备优化，主要是针对一些老的不识别viewport的浏览器，比如黑 -->
    <meta name="HandheldFriendly" content="true">
    <!-- 微软的老式浏览 -->
    <meta name="MobileOptimized" content="320">
    <!-- uc强制竖屏 -->
    <!-- <meta name="screen-orientation" content="portrait"> -->
    <!-- QQ强制竖屏 -->
    <!-- <meta name="x5-orientation" content="portrait"> -->
    <!-- 不能旋转，问题出在这 -->
    <!-- UC强制全屏 -->
    <meta name="full-screen" content="yes">
    <!-- QQ强制全屏 -->
    <meta name="x5-fullscreen" content="true">
    <!-- UC应用模式 -->
    <meta name="browsermode" content="application">
    <!-- QQ应用模式 -->
    <meta name="x5-page-mode" content="app">
    <!-- windows phone  -->
    <meta name="msapplication-tap-highlight" content="no">
    <!-- Load the polyfill to switch-hit between Chrome and Firefox -->
    <style>
        * {
            margin: 0;
            padding: 0;
        }

        html, body {
            height: 100%;
            width: 100%;
            margin: 0;
            padding: 0;
```

```
        background-color: grey;
    }
</style>
<!-- 云手机对接sdk,请联系项目经理获取最新版本 -->
<script type="text/javascript" src="js/BgsSdk.min.1.7.4.js"></script>
<script type="text/javascript" src="https://cdn.bootcdn.net/ajax/libs/axios/0.26.1/axios.min.js"></script>

<script>
    var uuid = 'test123'; // 有应用来定义，主要用来区分用户
    const getServerTokenUrl = 'https://xxxx/getServerToken'; // 后台服务可以参考后台demo 方案
    var serverToken = ''; // 从业务后台获取
    async function getServerToken() {
        const res = await axios.post(getServerTokenUrl, {
            uuid: uuid
        });
        if(res.code !== 0) {
            console.log('getServerToken fail res:' + JSON.stringify(res));
            return;
        }
        serverToken = res.data.serverToken;
        // 开始链接
        BgsSdk.startPhone(serverToken);
    }
    function startPhone() {
        // 初始化参数
        var params = {
            viewId: 'phoneBox', // 必填
            uuid: uuid, // 必填
            callbacks: {
                // 初始化失败回调
                onInitFail: function(code,msg) {
                    console.log('on init code:' + code + ' msg:' + msg);
                },
                onInitSuccess: function() {
                    console.log('on init success:');
                },
                // 链接成功回调
                onConnectSuccess: function () {
                    console.log('on success');
                },
                // 链接失败回调
                onConnectFail: function (code, msg) {
                    console.log('on fail code:' + code);
                    stopPhone();
                },
                // 结束回调
                onStoped: async function() {
                    // 已断开云手机
                    console.log('on stoped');
                }
            }
        }
        // 开始初始化
        let initRet = BgsSdk.initPhone(params);
        if (initRet) {
            // 获取serverToken
            getServerToken();
        }

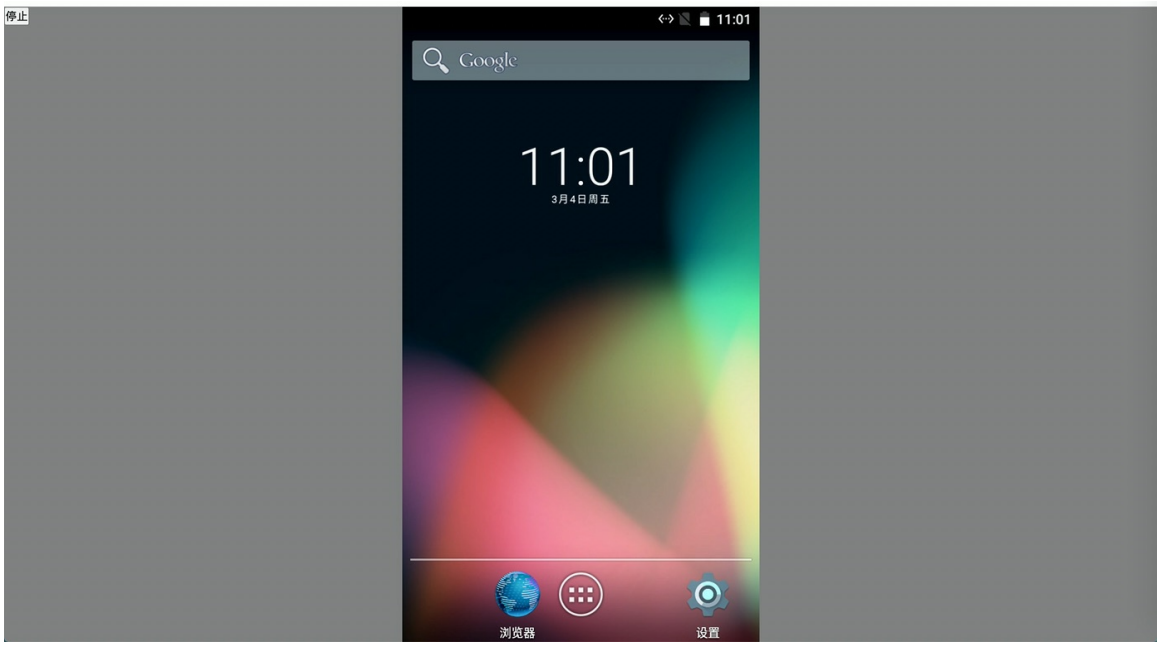
    }
    // 结束试玩
    function stopPhone() {
        BgsSdk.stopPhone();
```

```
            BgsSdk.stopPhone();
        }

    </script>
</head>

<body onload="startPhone();">
<input type="button" value="停止" onclick="stopPhone()" style="position: fixed;top: 0; left: 0;">
    <div class="phoneBox" id="phoneBox" style="background-color: grey;width: 100%;height: 100%">
    </div>
</body>
</html>
```

4.4 运行效果



5 获取更多帮助文档

接入云手机服务后，进入云手机管理控制台，点击控制台右上角“帮助文档”解锁更多

相关协议&政策

服务等级协议SLA

服务水平协议SLA

协议生效时间：2024年10月24日

本服务等级协议（Service Level Agreement，以下简称 "SLA"）规定了百度智能云向客户提供的百度云手机服务可用性等级指标及赔偿方案。

1. 服务可用性定义

1.1 服务周期：一个服务周期为一个自然月

1.2 服务周期总分钟数：服务周期内的总天数×24（小时）×60（分钟）

1.3 服务不可用分钟数：当某一分钟内，甲方所有尝试与指定的云手机实例建立连接的连续请求均失败，则视为该分钟内该云手机实例服务不可用。在一个服务周期内，单个云手机实例的不可用分钟数为其所有不可用分钟数的总和。

1.4 服务可用性计算公式：以单台云手机实例为维度，服务可用性=【（服务周期总分钟数-服务不可用分钟数）/服务周期总分钟数】*100%

2. 服务可用性

2.2 服务可用性承诺 2.1 对于单个云手机实例维度，乙方承诺一个服务周期内云手机实例的服务可用性不低于99%；

2.2 如云手机实例服务未达到上述服务可用性承诺，甲方可以根据约定获得赔偿。赔偿范围不包括以下原因所导致的不可用：

1）乙方预先通知甲方后进行系统维护所引起的，包括割接、维修、升级和模拟故障演练；

2）任何乙方所属设备以外的网络、设备故障或配置调整引起的；

3）甲方的应用程序或数据信息受到黑客攻击而引起的；

4）甲方维护不当或保密不当致使数据、口令、密码等丢失或泄漏所引起的；

5）甲方的疏忽或由甲方授权的操作所引起的；

6）不可抗力以及意外事件引起的；

7）未达到服务可用性承诺的云手机实例只赔付不高于同等数量的云手机实例服务自身，不赔付云手机服务所关联的其他云服务；

8）其他非乙方原因所造成的云手机实例服务无法正常使用。

3. 服务赔偿条款

3.1. 赔偿标准 1）对于单个云手机实例，可按照下表中的标准获得赔偿，赔偿方式仅限于抵扣结算费用且赔偿总额不超过未达到服务可用性承诺当月客户甲方就该云手机实例支付的当月单个云手机实例月度服务费。

服务可用性（%）	**赔付时长（小时）**
等于或高于99%	不赔付
低于99%但等于或高于95%	(99%-服务可用性)*月度自然日天数*24小时*2
低于95%	(99%-服务可用性)*月度自然日天数*24小时*3

3.2. 赔偿申请时限 甲方可以在每月收到账单后3个工作日内提出。超出申请时限的赔偿申请将不被受理，甲方有客观合理事由延期申请赔偿的除外。

百度云手机软件开发工具包隐私政策

百度云手机软件开发工具包隐私政策

欢迎使用百度云手机软件开发工具包（SDK）（简称“百度云手机SDK”）服务！

百度云手机SDK 是一款为移动应用开发者（以下简称“开发者”）提供百度云手机仿真务的软件开发工具包（SDK）。开发者在其移动应用内集成百度云手机SDK后，可通过百度云手机SDK平台向其移动应用（以下简称“开发者应用”）的最终用户（以下简称“最终用户”）提供本隐私政策所说明的功能及服务。 此隐私政策旨在帮助开发者及最终用户了解我们收集最终用户个人信息的类型及我们如何利用和保护最终用户的个人信息。为了便于开发者及最终用户阅读及理解，我们将专门术语进行了定义，请参见本隐私政策“附录1：名词解释”来了解这些定义的具体内容。

特别说明：

1、如果开发者在其移动应用中集成并使用百度云手机SDK服务，则开发者应承诺：

（1）开发者应遵守收集、使用最终用户个人信息有关的所有可适用法律、政策和法规，保护用户个人信息安全。

（2）开发者应将在其移动应用中集成并使用百度云手机SDK服务的情况，以及百度云手机SDK对最终用户必要个人信息的收集、使用和保护规则（即本隐私政策），在其移动应用的显著位置或以其他可触达最终用户的方式告知最终用户（包括但不限

于：在其移动应用隐私政策显眼处提供最终用户可浏览本隐私政策的链接），并获得最终用户对于百度云手机SDK收集、使用最终用户相关个人信息的完整、合法、在使用百度云手机SDK服务期间持续有效的授权同意。如果开发者的移动应用最终用户是未满14周岁的未成年人，请开发者务必确保获得最终用户的父母或其他监护人对于百度云手机SDK收集、使用最终用户相关个人信息的完整、合法、在使用百度云手机SDK服务期间持续有效的授权同意。

（3）开发者应向最终用户提供易于操作的查阅、更正、补充、删除、复制或转移其个人信息，撤回或更改其授权同意，注销其个人账号，要求开发者就个人信息处理规则作出解释说明等用户权利实现机制。

2、我们希望集成并使用百度云手机SDK服务的开发者应用以合法合规的方式收集、使用最终用户的个人信息，但我们并不了解且无法控制任何开发者以及他们的移动应用如何使用他们所控制的最终用户个人信息，因此也不应为其行为负责。我们建议最终用户在认真阅读开发者应用相关隐私政策，在确认充分了解并同意他们如何收集、使用最终用户的个人信息后再使用开发者应用。

3、本隐私政策不适用于展示在、链接到或再封装我们的服务的那些适用第三方隐私政策、并由第三方提供的服务。虽然第三方展示在、链接到或再封装我们的服务，但我们并不了解或控制其行为，因此也不为其行为负责。请开发者及最终用户在已查看并接受其隐私政策之前，谨慎访问或使用其服务。

4. 最终用户具体获得的百度云手机SDK服务内容由开发者根据其移动应用需要进行选择，可能因为最终用户所使用的开发者应用不同而有所差异，百度云手机SDK可能获得的个人信息取决于最终用户所使用的开发者应用的具体类型/版本以及最终用户所使用的功能。如果在部分开发者应用版本中不涵盖某些服务内容或未提供特定功能，则本隐私政策中涉及到上述服务/功能及相关个人信息的内容将不适用。

请开发者及最终用户务必认真阅读本隐私政策，在确认充分了解并同意后再集成并使用百度云手机SDK服务。

本隐私政策将帮助开发者及最终用户了解以下内容：

1. 我们如何收集和使用最终用户的个人信息
2. 我们如何使用 Cookie 和同类技术
3. 我们如何共享、转让、公开披露最终用户的个人信息
4. 我们如何保存及保护最终用户的个人信息
5. 我们如何保障最终用户的个人信息相关权利行使
6. 我们如何处理未成年人的个人信息
7. 隐私政策的修订
8. 如何联系我们

我们珍视最终用户在向我们提供最终用户个人信息时对我们的信任，我们将按照本隐私政策处理最终用户的个人信息并保障最终用户信息的安全。

一、我们如何收集和使用最终用户的个人信息

（一）为帮助开发者向最终用户提供相应功能及服务

为了帮助开发者向最终用户提供相应功能及服务，当最终用户使用相应功能及服务时，我们会通过开发者应用向系统申请最终用户设备的相应权限。开发者应确保最终用户可以随时通过取消系统授权开发者应用获取相应设备权限或其他开发者应用提供的授权设置，停止我们对最终用户个人信息的收集，之后最终用户可能将无法使用基于相应个人信息而提供的相关服务或功能，或者无法达到基于相应个人信息提供的相关服务拟达到的效果，但不会影响最终用户正常使用百度云手机SDK的其他不基于相应个人信息即可实现的业务功能。

1.各项功能及服务涉及的个人信息

序号	功能及服务	个人信息类型	收集方式	适用系统版本
1	为帮助开发者向最终用户提供应用服务功能，以便准确分析存在问题	设备信息、位置信息、传感器信息	SDK直接采集	iOS及Android
2	为帮助开发者向最终用户提供登录功能	剪切板中的口令、链接等内容	SDK直接采集	iOS及Android

位置为敏感个人信息，最终用户可以随时通过取消系统授权开发者应用获取相应设备权限或其他开发者应用提供的授权设置停止我们对上述敏感个人信息的收集，之后最终用户可能将无法使用基于上述敏感个人信息而提供的相关服务或功能，或者无法达到基于上述敏感个人信息提供的相关服务拟达到的效果，但不会影响最终用户正常使用百度云手机SDK的其他不基于上述敏感个人信息即可实现的业务功能。

2. 设备权限调用

为了保证最终用户能正常使用百度云手机SDK相应功能及服务，我们会通过开发者应用向系统申请最终用户设备的以下系统设备权限，申请前我们会征询最终用户的同意，最终用户可以选择“允许”或“禁止”权限申请。经过最终用户的授权后我们会开启相关权限，最终用户可以随时在系统中取消授权，最终用户取消授权会导致最终用户无法使用相关的业务功能，但不会导致最终用户无法使用其他业务功能。各项功能及功能对设备权限的调用情况如下：

Android & IOS系统版本

设备权限	功能及服务	权限授权方式
读取/写入外部存储卡	升级程序下载、图片保存、错误日志保存	**授权方式由设备系统开发方以及开发者应用决定；当最终用户同意向开发者应用授予该权限时开启**
读取/写入外部存储卡	升级程序下载、图片保存、错误日志保存	
网络	升云端功能应用及服务	
获取位置	错误日志分析	
访问电话状态（部分机型也称为“设备信息”权限）	错误日志分析	
摄像头	使用百度云手机进行拍摄	
录音	使用百度云手机进行录音	
安装应用	检查到有更新包时下载安装	
震动	操作百度云手机时提示用户	
读取日志	百度云手机收集异常信息上报	
检索正在运行的应用	百度云手机客服判断当前页是否是在最上层	

在不同设备中，权限显示方式及关闭方式可能有所不同，具体请最终用户参考设备及系统开发方说明或指引。

（二）保证服务安全、优化和改善服务目的

为了帮助开发者向最终用户提供上述功能及服务，同时为了更准确定位并解决开发者以及最终用户在使用百度云手机SDK产品和服务时遇到的问题，改进及优化百度云手机SDK产品和服务在开发者侧以及最终用户侧的双重体验，更准确定位并解决最终用户在使用百度云手机SDK服务时遇到的问题，改进及优化百度云手机SDK的服务体验，提高百度云手机SDK服务的安全性，预防、发现、调查欺诈、危害安全、非法或违反与我们的协议、政策或规则的行为，以保护开发者、最终用户、我们或我们的关联公司、合作伙伴及社会公众的合法权益，我们会收集最终用户的**设备信息、位置信息、日志信息及其他与登录环境相关的信息。**

（三）个人信息的匿名化处理

在不公开披露、对外提供最终用户个人信息的前提下，百度公司有权对匿名化处理后的用户数据库进行挖掘、分析和利用（包括商业性使用），有权对产品/服务使用情况进行统计并与公众/第三方共享匿名化处理后的统计信息。

（四）事先征得授权同意的例外

请注意，在以下情形中，收集、使用个人信息无需事先征得最终用户的授权同意：

1. 与国家安全、国防安全直接相关的；
2. 为订立、履行个人作为一方当事人的合同所必需；
3. 为履行法定职责或者法定义务所必需；
4. 为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
5. 与犯罪侦查、起诉、审判和判决执行等直接有关的；
6. 出于维护最终用户或其他个人的生命、财产等重大合法权益但又很难得到本人同意的；
7. 依照法律法规的规定在合理的范围内收集最终用户自行向社会公众公开或其他已经合法公开的个人信息；
8. 依照法律法规的规定在合理的范围内从合法公开披露的信息中收集最终用户的个人信息，如合法的新闻报道、政府信息公开等渠道；
9. 为公共利益实施新闻报道、舆论监督等行为，在合理的范围内处理个人信息；
10. 学术研究机构基于公共利益开展统计或学术研究所必要，且对外提供学术研究或描述的结果时，对结果中所包含的个人信息进行去标识化处理的；
11. 法律法规规定的其他情形。

提示最终用户注意，当我们要将信息用于本隐私政策未载明的其它用途时，会事先征求最终用户的同意。

二、我们如何使用 Cookie 和同类技术

Cookie是支持服务器端（或者脚本）在客户端上存储和检索信息的一种机制。当最终用户使用百度云手机SDK产品或服务时，我们会向最终用户的设备发送一个或多个Cookie或匿名标识符。当最终用户与百度云手机SDK服务进行交互时，我们允许Cookie或者匿名标识符发送给百度公司服务器。Cookie 通常包含标识符、站点名称以及一些号码和字符。运用Cookie技术，百度公司能够了解最终用户的使用习惯，记住最终用户的偏好，省去最终用户输入重复信息的步骤，为最终用户提供更加周到的个性化服务，或帮最终用户判断最终用户账户的安全性。Cookie还可以帮助我们统计流量信息，分析页面设计和广告的有效性。

我们不会将 Cookie 用于本政策所述目的之外的任何用途。最终用户可根据自己的偏好管理或删除 Cookie。有关详情，请参见 AboutCookies.org。最终用户可以清除计算机上保存的所有 Cookie，大部分网络浏览器都设有阻止 Cookie 的功能。但如果最终用户这么做，则需要在每一次访问我们的网站时亲自更改用户设置，但最终用户可能因为该等修改，无法登录或使用依赖于Cookie的百度公司提供的服务或功能。

三、我们如何共享、转让、公开披露最终用户的个人信息

（一）共享

除非经过您本人事先单独同意或符合其他法律法规规定的情形，我们不会向除百度公司以外的第三方共享您的个人信息，但经过处理无法识别特定个人且不能复原的除外。

对我们与之共享个人信息的公司、组织和个人，我们会对其数据安全环境进行调查，与其签署严格的保密协定，要求他们按照依法采取保密和安全措施来处理个人信息。

1. 在下列情况下，经过最终用户的授权同意，我们可能会共享的个人信息： 仅为实现本隐私政策中声明的目的，我们的某些服务将由授权合作伙伴提供。我们可能会与合作伙伴共享最终用户的某些个人信息，以提供更好的客户服务和用户体验。我们

仅会出于合法、正当、必要、特定、明确的目的共享最终用户的个人信息，并且只会共享与提供服务相关的个人信息。我们的合作伙伴无权将共享的个人信息用于任何其他用途。

目前，我们的授权合作伙伴包括以下类型：

(1) 服务平台或服务提供商。百度各产品接入了丰富的第三方服务。当最终用户选择使用该第三方服务时，最终用户授权我们将该信息提供给第三方服务平台或服务提供商，以便其基于相关信息为最终用户提供服务。

(2) 软硬件/系统服务提供商。当第三方软硬件/系统产品或服务与百度的产品或服务结合为最终用户提供服务时，经最终用户授权，我们会向第三方软硬件/系统服务提供商提供最终用户必要的个人信息，以便最终用户使用服务，或用于我们分析产品和服务使用情况，来提升最终用户的使用体验。

(3) 广告、咨询类服务商/广告主。未经最终用户授权，我们不会将最终用户的个人信息与提供广告、咨询类服务商共享。但我们可能会将经处理无法识别最终用户的身份且接收方无法复原的信息，例如经匿名化处理的用户画像，与广告或咨询类服务商或广告主共享，以帮助其在不识别最终用户个人的前提下，提升广告有效触达率，以及分析我们的产品和服务使用情况等。

2. 对我们与之共享个人信息的公司、组织和个人，我们会对其数据安全环境进行调查，与其签署严格的保密协定，要求他们按照我们的说明、本隐私政策以及其他任何相关的保密和安全措施来处理个人信息。

（二）转让

我们不会将最终用户的个人信息转让给除关联公司外的任何公司、组织和个人，但以下情况除外：

1. 事先获得最终用户的明确授权或同意；
2. 满足法律法规、法律程序的要求或强制性的政府要求或司法裁定；
3. 如果我们或我们的**关联公司**涉及合并、分立、清算、资产或业务的收购或出售等交易，最终用户的个人信息有可能作为此类交易的一部分而被转移，我们将确保该等信息在转移时的机密性，并尽最大可能确保新的持有最终用户个人信息的公司、组织继续受此隐私政策的约束，否则我们将要求该公司、组织重新向最终用户征求授权同意。

（三）公开披露

我们仅会在以下情况下，公开披露最终用户的个人信息：

1. 获得最终用户的单独同意；
2. 基于法律法规、法律程序、诉讼或政府主管部门强制性要求下。

（四）共享、转让、公开披露个人信息时事先征得授权同意的例外

在以下情形中，共享、转让、公开披露个人信息无需事先征得最终用户的授权同意：

1. 与国家安全、国防安全直接相关的；
2. 为订立、履行个人作为一方当事人的合同所必需；
3. 为履行法定职责或者法定义务所必需；
4. 与公共安全、公共卫生、重大公共利益直接相关的。例如：为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
5. 与犯罪侦查、起诉、审判和判决执行等直接相关的；
6. 出于维护个人信息主体或其他个人的生命、财产等重大合法权益但又很难得到本人同意的；
7. 依照法律法规的规定在合理的范围内处理个人自行公开或者其他已经合法公开的个人信息，例如：合法的新闻报道、政府信息公开等渠道等；
8. 法律、行政法规另有规定的情形。

四、我们如何保存及保护最终用户的个人信息

（一）保存期限

我们将在百度云手机SDK自身提供服务以及开发者应用提供服务所需的期限内保存您的个人信息，但法律法规对保存期限另有规定、您同意留存更长的期限、保证服务的安全与质量、实现争议解决目的、技术上难以实现等情况下，在前述保存期限到期后，我们将依法、依约或在合理范围内延长保存期限。

在超出保存期限后，我们将根据法律规定删除您的个人信息或进行匿名化处理。

（二）保存地域

原则上，我们在中华人民共和国境内收集和产生的个人信息，将存储在中华人民共和国境内。如您的个人信息可能会被转移到您使用产品或服务所在国家/地区的境外管辖区，或者受到来自这些管辖区的访问，我们会严格履行法律法规规定的义务并按照法律规定事先获得您的单独同意。此类管辖区可能设有不同的数据保护法，甚至未设立相关法律。在此类情况下，我们会按照中国现行法律的规定传输您的个人信息，并会确保您的个人信息得到在中华人民共和国境内足够同等的保护。例如，我们会请求您对跨境转移个人信息的同意，或者在跨境数据转移之前实施数据去标识化等安全举措。

（三）安全措施

1. 我们会以“最小化”原则收集、使用、存储和传输用户信息，并通过用户协议和隐私政策告知您相关信息的使用目的和范围。

2. 我们非常重视信息安全。我们成立了专责团队负责研发和应用多种安全技术和程序等，我们会对安全管理负责人和关键安全岗位的人员进行安全背景审查，我们建立了完善的信息安全管理制度和内部安全事件处置机制等。我们会采取适当的符合业界标准的安全措施和技术手段存储和保护您的个人信息，以防止您的信息丢失、遭到被未经授权的访问、公开披露、使用、毁损、丢失或泄漏。我们会采取一切合理可行的措施，保护您的个人信息。我们会使用加密技术确保数据的保密性；我们会使用受信赖的保护机制防止数据遭到恶意攻击。

3. 我们会对员工进行数据安全的意识培养和安全能力的培训和考核，加强员工对于保护个人信息重要性的认识。我们会对处理个人信息的员工进行身份认证及权限控制，并会与接触您个人信息的员工、合作伙伴签署保密协议，明确岗位职责及行为准则，确保只有授权人员才可访问个人信息。**若有违反保密协议的行为，会被立即终止与百度公司的合作关系**，并会被追究相关法律责任，对接触个人信息人员在离岗时也提出了保密要求。

4. 我们提醒您注意，互联网并非绝对安全的环境，**当您通过百度云手机SDK中嵌入的第三方社交软件、电子邮件、短信等与其他用户交互您的地理位置或行踪轨迹信息时，不确定第三方软件对信息的传递是否完全加密，注意确保您个人信息的安全。**

5. 我们也请您理解，在互联网行业由于技术的限制和飞速发展以及可能存在的各种恶意攻击手段，即便我们竭尽所能加强安全措施，也不可能始终保证信息的百分之百安全。**请您了解，您使用我们的产品和/或服务时所用的系统和通讯网络，有可能在我们控制之外的其他环节而出现安全问题。**

6. **根据我们的安全管理制度，个人信息泄露、毁损或丢失事件被列为最特大安全事件，一经发生将启动公司最高级别的紧急预案**，由安全部、公共事务部、法务部等多个部门组成联合应急响应小组处理。

（四）安全事件通知

1. 我们会制定网络安全事件应急预案，及时处置系统漏洞、计算机病毒、网络攻击、网络侵入等安全风险，在发生危害网络安全的事件时，我们会立即启动应急预案，采取相应的补救措施，并按照规定向有关主管部门报告。

2. 个人信息泄露、毁损、丢失属于公司级特大安全事件，我们会负责定期组织工作组成员进行安全预案演练，防止此类安全事件发生。若一旦不幸发生，我们将按照最高优先级启动应急预案，组成紧急应急小组，在最短时间内追溯原因并减少损失。

3. 在不幸发生个人信息安全事件后，我们将按照法律法规的要求，及时向您告知安全事件的基本情况和可能的影响、我们已采取或将要采取的处理措施、您可自主防范和降低的风险的建议、对您的补救措施等。我们将及时将事件相关情况以站内通知、短信通知、电话、邮件等您预留的联系方式告知您，难以逐一告知时我们会采取合理、有效的方式发布公告。同时，我们还将按照监管部门要求，主动上报个人信息安全事件的处置情况。请您理解，根据法律法规的规定，如果我们采取的措施能够有效避免信息泄露、篡改、丢失造成危害的，除非监管部门要求向您通知，我们可以选择不向您通知该个人信息安全事件。

五、我们如何处理未成年人的个人信息

百度公司非常重视对未成年人信息的保护。

百度云手机SDK的产品和服务主要面向成年人。如果最终用户是未满14周岁的未成年人，请务必在使用已集成百度云手机SDK的开发者应用前，在父母或其他监护人监护、指导下共同仔细阅读开发者应用隐私政策、本隐私政策以及《百度儿童个人信息保护声明》，并在征得监护人同意的前提下使用开发者应用或提供个人信息。

如果我们发现在未事先获得可证实的父母同意的情况下收集了未成年人的个人信息，将会采取措施尽快删除相关信息。

如果任何时候监护人有理由相信我们在未获监护人同意的情况下收集了未成年人的个人信息，请通过官方渠道联系我们，我们会采取措施尽快删除相关数据。

六、我们如何保障最终用户的个人信息相关权利行使

按照中国相关的法律、法规、标准，以及其他国家、地区的通行做法，我们将尽力协调、支持并保障最终用户对自己的个人信息行使以下权利：

1. 查阅权、更正及补充权、复制权、帐号注销权

鉴于最终用户通过开发者应用使用百度云手机SDK服务，并且使用服务时并不会注册/登录百度帐号，为保障最终用户查阅、更正、补充、复制其个人信息以及注销其开发者应用帐号的权利实现，我们在本隐私政策以及其他与开发者的约定中，要求开发者承诺提供便于操作的实现方式。如最终用户要查阅、更正、补充、复制其个人信息或注销其开发者应用帐号，应通过开发者提供的上述权利实现方式，如开发者未按照承诺进行提供，最终用户可通过八、如何联系我们中的方式与我们取得联系，我们将尽力协调、支持并保障最终用户的上述权利实现。

2. 删除权

为保障最终用户删除其个人信息的权利实现，我们在本隐私政策以及其他与开发者的约定中，要求开发者承诺提供便于操作的实现方式。如最终用户要删除其个人信息，应通过开发者提供的上述权利实现方式，如开发者未按照承诺进行提供，在以下情形中，最终用户可以通过八、如何联系我们中的方式与我们取得联系，向我们提出删除个人信息的请求：

1. 如果我们违反法律法规或与最终用户的约定处理其个人信息；
2. 如果我们的处理目的已实现、无法实现或者为实现处理目的不再必要；
3. 如果我们停止提供产品或者服务，或者保存期限已届满；
4. 如果最终用户撤回同意；
5. 法律、行政法规规定的其他情形。 当最终用户从我们的服务中删除信息后，我们可能不会立即在备份系统中删除相应的信息，但会在备份更新时删除这些信息。请最终用户知晓并理解，如果法律、行政法规规定的或本隐私政策说明的保存期限未届满，或者删除个人信息从技术上难以实现的，我们会停止除存储和采取必要的安全保护措施之外的处理。

3. 撤回同意

每个业务功能需要一些基本的个人信息才能得以完成。对于额外收集的个人信息的收集和使用，最终用户可以随时给予或收回其授权同意。

最终用户可以在设备系统中直接关闭本隐私政策说明的我们可能调用的设备系统权限，或开发者应用提供的其他授权设置（如适用），改变同意范围或撤回其授权。

当最终用户撤回同意后，我们无法继续为最终用户提供撤回同意所对应的服务，也将不再使用最终用户相应的个人信息。但最终用户撤回同意的决定，不会影响此前基于其同意而开展的个人信息处理。

4. 可携带权

为保障最终用户转移其个人信息的权利实现，我们在本隐私政策以及其他与开发者的约定中，要求开发者承诺提供便于操作的实现方式。如最终用户要转移其个人信息，应通过开发者提供的上述权利实现方式，如开发者未按照承诺进行提供，最终用户可以通过八、如何联系我们中的方式与我们取得联系，我们将尽力协调、支持并保障其上述权利实现。

在法律法规规定的条件下，同时符合国家网信部门规定指令和条件的，如果技术可行，最终用户也可以要求我们将其个人信息转移至最终用户指定的其他主体。

5. 提前获知产品和服务停止运营权。

百度云手机SDK愿一直陪伴开发者和最终用户，若因特殊原因导致百度云手机SDK产品停止运营，我们将在合理期间内在产品或服务的主页面或站内信或向开发者和最终用户发送电子邮件或其他合适的能触达其方式通知开发者和最终用户，并将停止对最终用户个人信息的收集，同时会按照法律规定对已收集的最终用户的个人信息进行删除或匿名化处理。

6. 获得解释的权利

最终用户有权要求我们就个人信息处理规则作出解释说明。最终用户可以通过八、如何联系我们中的方式与我们取得联系。

七、隐私政策的修订与通知

我们的隐私政策可能变更。

未经最终用户明确同意，我们不会削减最终用户按照本隐私政策所应享有的权利。我们会在本页面上发布对本隐私政策所做的任何变更。

对于重大变更，**我们会在百度云手机SDK官方网站的主要曝光页面向开发者以及最终用户公示，并以任何可触达的方式通知开发者。若开发者不同意该等变更可以停止集成并使用百度云手机SDK产品和服务，若开发者继续集成并使用百度云手机SDK产品和/或服务，即表示开发者同意受修订后的本隐私政策的约束，并将此变更通知最终用户，获取最终用户对此变更的完整、合法、在使用百度云手机SDK服务期间持续有效的授权同意**。若最终用户不同意该等变更，可以停止使用百度云手机SDK产品和服务，开发者应向用户提供相应实现机制；若最终用户继续使用百度云手机SDK产品和/或服务，即表示最终用户同意受修订后的本隐私政策的约束。

本政策所指的重大变更包括但不限于：

1. 我们的服务模式发生重大变化。如处理个人信息的目的、处理的个人信息类型、个人信息的使用方式等；

2. 我们在所有权结构、组织架构等方面发生重大变化。如业务调整、破产并购等引起的所有者变更等；

3. 个人信息共享、转让或公开披露的主要对象发生变化；

4. 最终用户参与个人信息处理方面的权利及其行使方式发生重大变化；

5. 我们负责处理个人信息安全的责任部门、联络方式及投诉渠道发生变化时；

6. 个人信息安全影响评估报告表明存在高风险时。

本政策更新后，我们会将本政策的旧版本存档，供最终用户查阅。

如有本隐私政策未尽事宜，以《百度隐私权保护声明》为准。

八、如何联系我们

百度云手机SDK的成长离不开各方开发者及最终用户的共同努力，我们非常感谢开发者及最终用户对百度云手机SDK数据更新、使用反馈方面的贡献。

开发者及最终用户可以通过官方渠道反馈开发者及最终用户对百度云手机SDK产品和服务的建议以及在使用过程中遇到的问题，以帮助我们优化产品功能及服务，使更多用户更加便捷的使用我们的产品和服务。

开发者及最终用户可以通过个人信息保护问题反馈平台（http://help.baidu.com/personalinformation） 同我们联系。

开发者及最终用户也可以通过如下联络方式同我们联系：

中国北京市海淀区上地十街10号

北京百度网讯科技有限公司 法务部

邮政编码：100085

为保障我们高效处理开发者/最终用户的问题并及时向开发者/最终用户反馈，需要开发者/最终用户提交身份证明、有效联系方式和书面请求及相关证据，我们会在验证开发者/最终用户的身份后处理开发者/最终用户的请求。

如果开发者/最终用户对我们的回复不满意，特别是最终用户认为我们的个人信息处理行为损害了最终用户的合法权益，开发者/最终用户还可以通过以下外部途径寻求解决方案：**向北京市海淀区人民法院提起诉讼。**

附录1：名词解释

个人信息是指以电子或者其他方式记录的与已识别或者可识别的自然人有关的各种信息，不包括匿名化处理后的信息。个人信息包括姓名、出生日期、身份证件号码、个人生物识别信息、住址、通信通讯联系方式、通信记录和内容、帐号密码、财产信息、征信信息、行踪轨迹、住宿信息、健康生理信息、交易信息等。

敏感个人信息是指一旦泄露或者非法使用，容易导致自然人的人格尊严受到侵害或者人身、财产安全受到危害的个人信息，包括生物识别、宗教信仰、特定身份、医疗健康、金融账户、行踪轨迹等信息，以及不满十四周岁未成年人的个人信息。

设备是指可用于使用百度产品和/或服务的装置，例如桌面设备、平板电脑或智能手机。

设备信息可能包括最终用户用于安装、运行百度云手机SDK的终端设备的设备属性信息（例如最终用户的硬件型号，操作系统版本及系统情况，设备配置，国际移动设备身份码IMEI、国际移动用户识别码IMSI、网络设备硬件地址MAC、广告标识符IDFA、供应商标识符IDFV、移动设备识别码MEID、匿名设备标识符OAID 、集成电路卡识别码ICCID 、Android ID、硬件序列号等设备标识符）、设备连接信息（例如浏览器的类型、电信运营商、使用的语言、WIFI信息）以及设备状态信息（例如设备应用安装列表）。

日志信息是指我们的服务器所自动记录最终用户在访问百度云手机SDK时所发出的请求，例如最终用户的IP 地址、浏览器的类型和使用的语言、硬件设备信息、操作系统的版本、网络运营商的信息、最终用户访问服务的日期、时间、时长等最终用户在使用我们的产品或服务时提供、形成或留存的信息。

Cookie是指支持服务器端（或者脚本）在客户端上存储和检索信息的一种机制，通过增加简单、持续的客户端状态来扩展基于Web的客户端/服务器应用。服务器在向客户端返回HTTP对象的同时发送一条状态信息，并由客户端保存。状态信息中说明了该状态下有效的URL范围。此后，客户端发起的该范围内的HTTP请求都将把该状态信息的当前值从客户端返回给服务器，这个状态信息被称为Cookie。

位置信息是指通过GPS信息，WLAN接入点、蓝牙、基站以及其他传感器信息所获取的精确位置信息，也包括通过IP地址或其他网络信息等获取的粗略地理位置信息。

用户画像是指通过收集、汇聚、分析个人信息，对某特定自然人个人特征，如其职业、经济、健康、教育、个人喜好、信用、行为等方面做出分析或预测，形成其个人特征模型的过程。直接使用特定自然人的个人信息，形成该自然人的特征模型，称为直接用户画像。使用来源于特定自然人以外的个人信息，如其所在群体的数据，形成该自然人的特征模型，称为间接用户画像。

去标识化是指个人信息经过处理，使其在不借助额外信息的情况下无法识别特定自然人的过程。 匿名化是指通过对个人信息的技术处理，使得个人信息主体无法被识别，且处理后的信息不能被复原的过程。个人信息经匿名化处理后所得的信息不属于个人信息。

百度平台是指百度公司旗下各专门频道或平台服务（包括百度搜索、百度APP及衍生版、百度百科、百度知道、百度贴吧、百度智能云、百度手机助手及其他百度系产品https://www.baidu.com/more/） 等网站、程序、服务、工具及客户端。

关联公司是指百度云手机SDK的经营者北京百度网讯科技有限公司及其他与百度公司存在关联关系的公司的单称或合称。“关联关系”是指对于任何主体（包括个人、公司、合伙企业、组织或其他任何实体）而言，即其直接或间接控制的主体，或直接或间接控制其的主体，或直接或间接与其受同一主体控制的主体。前述“控制”指，通过持有表决权、合约或其他方式，直接或间接地拥有对相关主体的管理和决策作出指示或责成他人作出指示的权力或事实上构成实际控制的其他关系。

再次感谢开发者以及最终用户对百度云手机SDK的信任和使用！

北京百度网讯科技有限公司

2022年4月2日更新

2022年4月2日生效

常见问题

使用类

云手机的收费标准是什么？

详情请查看产品定价。 服务器整机售卖价格，欢迎提交工单联系客服，咨询专属福利。

可以在移动端使用云手机吗？

可以，客户需要自行将云手机的Android SDK或Web SDK集成到第三方APP上，即可在第三方APP上使用云手机。

支持连接的云手机实例规格有哪些？

目前有三种规格：
高配旗舰型：旗舰级ARM处理芯片；
中配普通型：主流级ARM处理芯片；
入门基础型：入门级ARM处理芯片。

云手机支持IOS系统吗？

支持，通过WebRtc投屏可以支持

退订管理问题

非产品不可用问题，不支持退订；

如遇到产品不可用问题（如云手机连接闪退、云手机系统不兼容应用等），购买后7天内可退订，费用以系统实际计费天数为准来收取，超过7天不支持退订。

请您在仔细了解并试用云手机后再进行购买。

客户常见问题

为什么不能安装、卸载、更新APP？

云手机处在连接状态上，默认有用户使用设备，不能安装、卸载、更新app，需要断开释放设备才可以进行以上操作

是否可以安装应用？在哪里安装应用？

可以，在控制台安装应用

为什么APP会安装失败？

APP最低兼容版本必须是安卓6.0,如果miniSdkVersion>23就不可以安装

云手机支持远程adb调用吗？

暂不支持adb，如有疑问请联系工单客服。

云手机可否支持绑定SIM卡？云手机能否收到登录短信？

暂不支持

一台电脑同时登录多台云手机，这些云手机的IP是否一样？

这些云手机虚拟IP不一样，每一台云手机都有自己独立的虚拟IP。

是否可以批量安装APP？

可以，控制台支持批量安装APP。

需要海外机房怎么办？

智能云云手机官网暂不支持，如有需要请联系工单客服。

云手机是否支持IP代理？

不支持，根据国家相关政策及有关部门规定，云手机不提供动态IP代理服务。