

# 设备大师高级函数接口

**版本: V2.3**

设备大师调用接口采用命令行的方式调用，使用前请先在云手机的配套服务中购买设备大师高级函数接口命令行格式如下：

- 1. `setphone` 该命令不带任何参数，表示从机型库中随机获取一款机型
- 2. `setphone -h` 带参数-h，获取自定义机型的命令行帮助
  - a: android id 设置 android id
  - m: manufacture 设置厂商
  - s: serialno 设置序列号
  - i: imei 设置 IMEI
  - w: wifi mac address 设置 WiFi MAC 地址
  - n: wifi name 设置 WiFi 名称
  - b: phone brand 设置云手机品牌
  - o: phone model 设置云手机型号
  - e: phone IMSI 设置 IMSI
  - p: phone product 设置云手机 product 参数
  - btname: bluetooth name 设置云手机 bluetooth name 参数
  - btaddr: bluetooth mac address 设置云手机 bluetooth mac address 参数
  - phoneno: phone number 设置云手机 phone number 参数
  - type: type 设置云手机 type 参数
  - user: user 设置云手机 user 参数
  - displayId: displayId 设置云手机 displayId 参数
  - baseVersion: baseband version 设置云手机基带版本号参数
  - operatorname: operatorname 设置云手机运营商参数
  - numeric: numeric 设置云手机运营商代码参数
  - country: country 设置云手机 country 参数
  - phonemac: wlan mac 设置云手机 wlan mac 参数，默认与 wifi mac 相同，单独设置后
  - osVersion: android release version 设置云手机 android 版本号参数
  - buildid: build id 设置值 ro.build.id 值
  - hardware: ro.hardware 设置 ro.hardware 值
  - simstate: sim state 设置 sim 卡状态，值为 READY 或者 ABSENT
  - incremental: incremental value 设置 ro.build.version.incremental 值
  - tags: tags 设置 ro.build.tags 值
  - host: host 设置 ro.build.host 值
  - codename: codename 设置 ro.build.version.codename 值
  - fingerprint: fingerprint 设置 ro.build.fingerprint 值
  - simserial: iccid 设置 iccid 值
  - nettype: nettype 设置 sim.data\_network\_type 值,如LTE
  - wifiiip: wifi ip 设置 wifi ip address 值
  - productname: ro.product.name 设置 ro.product.name 值
  - device: ro.product.device 设置 ro.product.device 值
  - board: ro.product.board 设置 ro.product.board 值
- 3. `setphone -a {androidid} -m {manufacture} -s {serialno} -i {imei} -w {wifiaddress} -n {wifiname} -b {brand} -o {model} -e {IMSI} -p {product}` 手动自定义设置云手机机型

- 4. `setphone backup 应用1 应用2 .....` 备份应用1, 应用2 (应用1, 应用2, 是包名, 如 `com.tencent.mobileqq`)
  - 返回值, 为备份的ID号, 如ID20190626163014, 该ID号需要自行保存, 再恢复备份的时候需要使用
- 5. `setphone restore {ID}` 恢复备份, ID为备份时返回的ID值
- 6. `setphone delete {ID}` 删除备份, ID为备份时返回的ID值
- 7. `setphone clearpkg {packagename}` 清理应用数据, `packagename` 位需要清理应用的包名, 会清理sdcard, 影响其他app, 功能与界面的强效清理相同, 如果不需清理 `sdcard`, 可自行调用 `pm clear {packagename}`, 该命令为系统标准命令, 清理 `/data/data` 目录, 只会影响清理的 `app`
- 8. `setphone phoneno {tel Number}` 设置本机号码接口, 例子参考脚本示例
- 9. `setphone dev {brand} {model}` 指定机型设置本机机型, 机型通过下面命令获取, 例子参考脚本示例
- 10. `setphone listdevice` 列出所有支持机型, 例子参考脚本示例
- 11. `setphone export {ID}` 导出备份, 导出目录为 `/sdcard/setcard/{ID}.zip`, 例子参考脚本示例
- 12. `setphone import {ID} {file}` 导入备份, 参数file的路径只能为全路径, 例子参考脚本示例
- 13. `setphone backupphone {NAME}` 机型数据备份, 不备份app数据
- 14. `setphone deletephone {NAME}` 删除机型数据备份
- 15. `setphone restorephone {NAME}` 恢复机型数据备份

## Lua 脚本调用示例:

`os.execute (" setphone ")` 随机从机型库设置机型。如需使用该功能, 请先在云手机的配套服务中购买设备大师服务。

```
os.execute (" setphone -a 8c1d1376b8291a32 -m KOPO -s OWKPELYSDZBGPFPU -i
864017031165699 -w 'c4:e2:b5:79:cd:25' -n TPLINK-ID9v -b KOPO -o KOPO_L9 -e
460024247497322 -p test ") 手动设置机型属性
```

`os.execute (" setphone backup com.tencent.mm > /sdcard/backup_result ")` 备份微信和机型, 并将ID导出到sdcard/backup\_result, 内容为ID20190626164035

```
os.execute (" setphone restore ID20190626164035 ") 恢复备份ID20190626164035
```

```
os.execute (" setphone delete ID20190626164035 ") 删除备份ID20190626164035
```

```
os.execute (" setphone phoneno 139xxxxxxx ") 设置本机号码
```

```
os.execute (" setphone dev lephone lephone_T7A ") 指定机型设置机型
```

```
os.execute (" setphone listdevice ") 列出支持的机型
```

```
os.execute (" setphone export ID20190626164035 ") 导出备份ID20190626164035
```

```
os.execute (" setphone import ID20190626164035
/sdcard/setcard/ID20190626164035.zip ") 导出备份
```

ID20190626164035, /sdcard/setcard/ID20190626164035.zip要求全路径

**note:**基础版、高级版和标准版不支持该接口更新