

DeviceMaster Advanced Function Interface

Version: V2.3

The DeviceMaster Advanced Function Interface is used through the shell command line. Before using it, please purchase the DeviceMaster advanced function interface in the supporting service of the cloud mobile phone. The command line format is as follows:

- 1. `setphone` this command does not take any parameters, which means that a model is randomly obtained from Server
- 2. `setphone -h` with the parameter -h, get the command line usage of the custom model
 - a: set `android id` value
 - m: set manufacturer value
 - s: set serial number
 - i: set `IMEI` value
 - w: set `WiFi MAC` address
 - n: set `WiFi` name
 - b: set cloud phone brand
 - o: set cloud phone model
 - e: set `IMSI` value
 - p: set cloud phone `product` parameter
 - btname: set cloud phone `bluetooth name` parameter
 - btaddr: set cloud phone `bluetooth mac address` parameter
 - phoneno: set cloud phone `phone number` parameter
 - type: set cloud phone `type` parameter
 - user: set cloud phone `user` parameter
 - displayId: set cloud phone `displayId` parameter
 - baseVersion: set cloud phone baseband version number parameter
 - operatorname: set cloud phone operator parameter
 - numeric: set cloud phone operator code parameter
 - country: set cloud phone `country` parameter
 - phonemac: set cloud phone `wlan mac` parameter
 - osVersion: set cloud phone `android` version number parameter
 - buildid: set `ro.build.id` value
 - hardware: set `ro.hardware` value
 - simstate: set `sim` card status, the value is `READY` or `ABSENT`
 - incremental: set `ro.build.version.incremental` value
 - tags: set `ro.build.tags` value
 - host: set `ro.build.host` value
 - codename: set `ro.build.version.codename` value
 - fingerprint: set `ro.build.fingerprint` value
 - simserial: set `iccid` value
 - nettype: set `sim.data_network_type` value, such as `LTE`
 - wifiip: set `wifi ip` address value
 - productname: set `ro.product.name` value
 - device: set `ro.product.device` value
 - board: set `ro.product.board` value
- 3. `setphone -a {androidid} -m {manifuature} -s {serialno} -i {imei} -w {wifiaddress} -n {wifiname} -b {brand} -o {model} -e {IMSI} -p {product}` manually set the cloud phone model
- 4. `setphone backup app1 app2 .....` backup application 1, backup application 2 (backup application 1, backup application 2, are packagename, such as com.tencent.mobileqq
*The return value is the ID number of the backup, such as ID20190626163014. This ID number needs to be saved by itself, and it needs to be used when restoring the backup
- 5. `setphone restore {ID}` restoring the backup, the ID is the ID value returned during the backup
- 6. `setphone delete {ID}` delete backup, ID is the ID value returned when backup
- 7. `setphone clearpkg {packagename}` clean up application data, the `packagename` bit needs to clear the package name of the application, it will clean up the sdcard and affect other apps, the function is the same as the powerful cleanup of the interface, if you don't need to clean up `sdcard`, you can call `pm clear {packagename}` by yourself, this command is a system standard command, clean up the `/data/data` directory, it will only affect the cleaned `app`
- 8. `setphone phoneno {tel Number}` set the local number interface, for an example, refer to the script example
- 9. `setphone dev {brand} {model}` specify the model to set the model of the machine. The model can be obtained by the following command. For an example, refer to the script example
- 10. `setphone listdevice` list all supported models, examples refer to script examples
- 11. `setphone export {ID}` export the backup, the export directory is `/sdcard/setcard/{ID}.zip`, for an example, refer to the script example
- 12. `setphone import {ID} {file}` to import a backup, the path of the parameter file can only be a full path. For an example, refer to the script example
- 13. `setphone backupphone {NAME}` model data backup, not app data backup
- 14. `setphone deletephone {NAME}` delete model data backup
- 15. `setphone restorephone {NAME}` restore model data backup

Example of `lua` script call:

```
os.execute(" setphone ") randomly set the model from the model library. If you want to use this function, please purchase the device master service in the supporting services of the cloud phone first.
os.execute(" setphone -a 8c1d1376b8291a32 -m KOPO -s 0WKPELYSDZBGFFPU -i 864017031165699 -w 'c4:e2:b5:79:cd:25' -n TPLINK-ID9v -b KOPO -o KOPO_L9 -e 460024247497322 -p test ") manually set model attributes
os.execute(" setphone backup com.tencent.mm > /sdcard/backup_result ") backup WeChat and model, and export the ID to /sdcard/backup_result, the content is ID20190626164035
os.execute(" setphone restore ID20190626164035 ") restore Backup ID 20190626164035
os.execute(" setphone delete ID20190626164035 ") delete backup ID 20190626164035
os.execute(" setphone phoneno 139xxxxxxx ") set up your phone number
os.execute(" setphone dev lephone lephone_T7A ") specify the model to set the model
os.execute(" setphone listdevice ") list supported models
os.execute(" setphone export ID20190626164035 ") export backup ID 20190626164035
os.execute(" setphone import ID20190626164035 /sdcard/setcard/ID20190626164035.zip ") export backup ID20190626164035, /sdcard/setcard/ID20190626164035.zip requires full path
```