Nástroj na extrakciu dát z Android zariadení

Martin Bažík

8. októbra 2017

Obsah

1 Úvod **2**
- 1.1 Požiadavky . . . 2
- 1.2 Moduly . . . 2

2 Inštalácia **4**
- 2.1 Stiahnutie potrebných nástrojov . . . 4

3 Spustenie **6**
- 3.1 Interaktívny mód . . . 6
  - 3.1.1 Možnosti . . . 6
  - 3.1.2 Beh nástroja . . . 6
- 3.2 Príkazový mód . . . 8
  - 3.2.1 Možnosti . . . 8
  - 3.2.2 Príkazy . . . 8
  - 3.2.3 Módy . . . 8
- 3.3 Výstup . . . 10

4 Moduly **13**
- 4.1 ADB . . . 13
- 4.2 Device . . . 14
  - 4.2.1 Device . . . 14
  - 4.2.2 App . . . 16
- 4.3 Cache . . . 16
  - 4.3.1 Schéma . . . 17

5 Potrebné dokončiť **18**

Kapitola 1

Úvod

Nástroj slúži na extrakciu dát z Android zariadení. Tento nástroj beží za pomoci adb (Android Debug Bridge), ktorý umožňuje spoluprácu s externým zariadením. Nástroj si v prvej fáze získa informácie o danom zariadení, aby sa s nimi mohlo následne pracovať a extrahovať špecifické dáta.

1.1 Požiadavky

- Operačný systém: Linux, Windows
- Softvér:
  - Android SDK (momentálne sa nachádza v Android Studio) dostupné na `https://developer.android.com/studio/index.html` - konkrétne nástroje adb a aapt
  - USB ovládače pre dané zariadenie, ak sa nenainštalujú automaticky
  - Python3
- Iné: USB kábel

1.2 Moduly

Nástroj sa skladá z niekoľkých častí. Každý modul obsahuje samostatnú časť implementácie a sú na seba naviazané.
Moduly:

- `xtractor.py` - hlavný modul, spúšťa celý nástroj v interaktívnom móde.
- `xtractor-cli.py` - hlavný modul, spúšťa nástroj v móde príkazového riadku.

- `adb.py` - modul predstavuje triedu ADB, ktorá sa stará of komunikáciu so zariadením využitím adb.
- `device.py` - modul obsahuje dve triedy - device a app. Objekty triedy device sú jednotlivé zariadenia, s ktorými sa komunikuje. Objekatami triedy app sú jednotlivé aplikácie dostupné na zariadení.
- `cache.py` -modul sa stará o ukladanie (cachovanie) prevzatých dát.

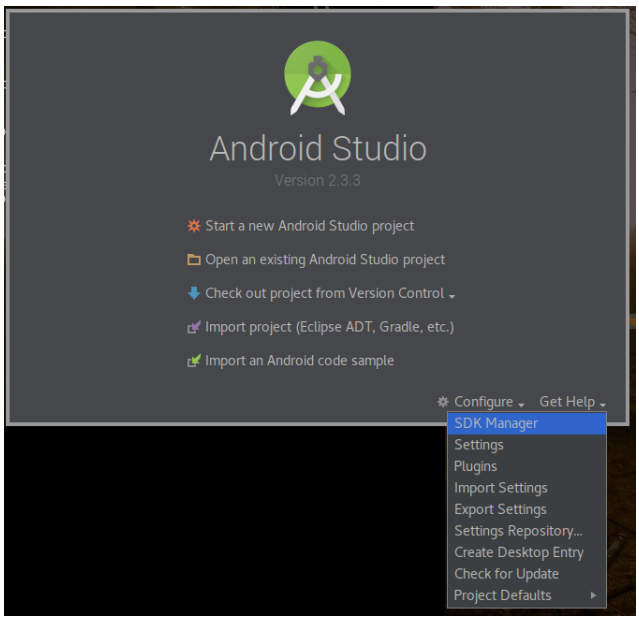
Kapitola 2

Inštalácia

2.1 Stiahnutie potrebných nástrojov

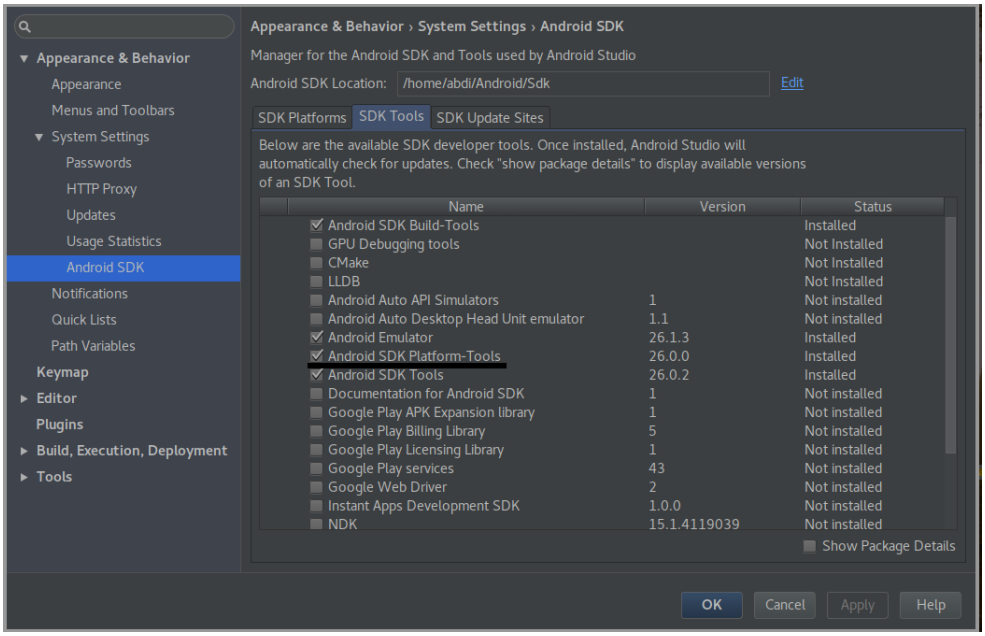
Android SDK slúži ako balíčkový manažér, ktorý umožňuje stiahnuť rôzne nástroje pre vývoj Android aplikácií. Patria medzi ne ADB, fastboot, aapt alebo AVD. Tieto nástroje sú rozdelené do niekoľkých balíkov. Pre tento nástroj sú dôležité balíčeky platform-tools a build-tools.
Postup pre Android Studio:

1. Spustite Android Studio a v nastaveniach vyberte `Configure` > `SDK Manager`



Obr. 2.1: Spustenie nástroja

2. Vyberte `SDK Tools` > `Android SDK Platform-Tools` > `OK`



Obr. 2.2: Stiahnutie balíčka

3. V prípade, že používate zariadenie od Google, stiahnite aj `Google USB Driver`.

Kapitola 3

Spustenie

Nástroj sa nachádza v zložke `script`. Je rozdelený na dve možné implemenacie. Prvou je interakivny mód, ktorá Vás prevedie celým procesom extrakcie. Druhou možnosťou je príkazový mód, ktorý vykoná iba daný príkaz a všetky metódy, ktoré sú s tým späté.

3.1 Interaktívny mód

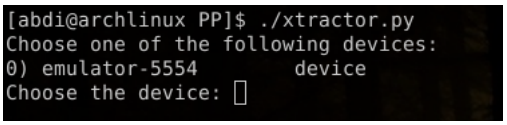
Nástroj sa nachádza v zložke `script`. Interaktívny mód sa púšťa pomocou spustiteľného súboru `xtractor.py`. Nástroj vás sám prevedie procesom extrakcie pomocou príkazov v interaktívnom móde.

3.1.1 Možnosti

`./xtractor.py -i <ip_adresa> -o <zlozka> -h`
`-i/--ip <ip_adresa>` - je možné použiť v prípade, že sa na zariadenie pripájame bezdrôtovo cez lokálnu wifi sieť.
`-o/--output <zlozka>` - zložka, do ktorej sa ukladajú extrahované súbory. V prípade, že nie je definovaná, sa lokálne vytvorí zložka `output/`.
`-h/--help` - pomoc

3.1.2 Beh nástroja

Po spustení sa zobrazia možnosti, s ktorými zariadeniami je možné komunikovať, viz obrázok 3.1.2.



Obr. 3.1: Spustenie nástroja

Voľba sa vyberá pomocou čísel na začiatku riadku. V tomto prípade je jedinou možnou voľbou 0. Po zvolení sa zo zariadenia posťahujú všetky dáta, ktoré nástroj potrebuje na ďalšie fungovanie.

Po stiahnutí všetkých potrebných dát zobrazí sa hláška `Done!` a nástroj požiada o zadanie príkazu nad týmto zariadením. Je ich možné rozdeliť na globálne a lokálne. Lokálne je možné použiť iba v prípade, že bola zvolená aplikácia pomocou príkazu `choose app` a aplikujú sa iba na dáta danej aplikácie. Globálne príkazy sa dostupné aj ak žiadna aplikácia nie je zvolená a sú aplikovateľné globálne na všetky dáta.

Možnými príkazmi sú:

- Globálne:
  - `help` - vypíše pomoc
  - `choose app` - slúži na výber aplikácie (stačí zadať čiastočný názov aplikácie)
  - `device info` - vypíše informácie o zariadení
  - `change device` - zmení zariadenie, s ktorým pracujeme
  - `list apps` - vypíše zoznam aplikácií aj s detailami
  - `sort apps` - zoradí aplikácie
  - `pull full` - stiahne vsetky súbory spojené s určitou aplikáciou
  - `exit` - ukončí program
- Lokálne:
  - `app info` - vypíše informácie o aplikácii
  - `list all files` - vypíše zoznam všetkých súborov na zariadení používaných niektorou aplikáciou
  - `list files` - vypíše vyfiltrovaný zoznam súborov (vypíšu sa až po zadaní tohoto príkazu)
  - `sort files` - zoradí súbory
  - `find files` - vyhľadá súbor
  - `time interval` - vyhľadá súbory modifikované v danom časovom intervale
  - `pull all` - stiahne všetky súbory na zariadení používaných vybranou aplikáciou
  - `pull filter` - stiahne vyfiltrované súbory
  - `print logs` - vytlačí logy spojené s danou aplikáciou
  - `save logs` - uloží logy spojené s danou aplikáciou
  - `reset filter` - zruší všetky filtre
  - `reset` - zruší výber aplikácie (nie je vybraná žiadna aplikácia)

3.2 Príkazový mód

Tento mód získa všetky potrebné zdroje a potom vykoná definovaný príkaz.

3.2.1 Možnosti

`./xtractor-cli.py -o <zlozka> -a <app_id> -d <id_zariadenia> -h`
`-o/--output <zlozka>` - zložka, do ktorej sa ukladajú extrahované súbory. V prípade, že nie je definovaná, sa lokálne vytvorí zložka `output/`.
`-a/--app <app_id>` - identifikátor aplikácie, s ktorou pracujeme
`-d/--device <id_zariadenia>` - identifikátor zariadenia, s ktorým pracujeme
`-h/--help` - pomoc
`-r/--record <id_zaznamu>` - identifikátor záznamu, s ktorým pracujeme / použitie namiesto parametru `-d`
`-f/--filter <filter>` - filter na následné filtrovanie dát

3.2.2 Príkazy

- `connect` - slúži na pripojenie zaradenia cez lokálnu wifi sieť
- `list-devices` - slúži na výpis pripojených zariadení
- `list-device-records` - slúži na výpis získaných záznamov
- Nasledovné príkazy vyžadujú zariadenie definované v parameteri -d alebo záznam v parametri -r:
  - `device-info` - slúži na prevzatie a výpis informácií o zariadení.
  - `find-applications` - slúži na prevzatie a výpis informácií ku aplikáciám. Vyhľadáva pre konkrétnu aplikáciu, ak je definovaná v parametri -a.
  - `find-files` - slúži na prevzatie a výpis informácií k súborom aplikácií. Vyhľadáva pre konkrétnu aplikáciu, ak je definovaná v parametri -a.
  - `find-logs` - slúži na prevzatie a výpis logov aplikácií. Vyhľadáva pre konkrétnu aplikáciu, ak je definovaná v parametri -a.
  - `pull` - slúži na prevzatie súborov spätých s aplikáciami. Vyhľadáva pre konkrétnu aplikáciu, ak je definovaná v parametri -a.

3.2.3 Módy

Existujú dva základné módy, v ktorých je možné aplikáciu spustiť. Aplikácia môže čerpať priamou extrakciou zo zariadenia alebo môže čerpať zo svojej cache, kde sa nachádzajú záznam skôr získaných dát. Okrem týchto módov, nástroj taktiež poskytuje možnosť filtrovania dát využitím cache databázy.

Extrakcia

Pre priamu extrakciu je potrebné mať zapojené zariadenie, z ktorého ideme čerpať dáta. Identifikátor pripojených zariadení je možné získať pomocou príkazu `./xtractor-cli.py -c list-devices`.

```
[abdi@archlinux PP]$ ./xtractor-cli.py -c list-devices
B6R4NZTKAQAMRW59:device
```

Obr. 3.2: Zariadenia

Následne je potrebné si vybrať požadovaný príkaz `[prikaz]` pre zariadenie `[zariadenie]` a spustiť nástroj:
`./xtractor-cli.py -d [zariadenie] -c [prikaz]`

Cache

Cache prináša možnosť prehľadávať údaje o predchádzajúcich extrakciách. Identifikátor záznamu extrakcie je možné získať pomocou príkazu:
`./xtractor-cli.py -c list-device-records`.

```
[abdi@archlinux PP]$ ./xtractor-cli.py -c list-device-records
1)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 15:59:02.397289
2)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 16:02:04.718943
3)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 16:04:42.681644
4)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 16:12:28.073686
5)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 16:14:19.126478
6)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 17:03:26.514578
7)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 20:40:53.182810
8)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 20:45:35.417893
9)      ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 21:08:52.976812
10)     ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 21:25:56.419355
11)     ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-06 22:05:22.258389
12)     ID: B6R4NZTKAQAMRW59    version: 5.1    architecture: arm       date: 2017-10-07 17:16:39.786835
```

Obr. 3.3: Záznamy

Následne je potrebné si vybrať požadovaný príkaz `[prikaz]` pre záznam `[zaznam]` a spustiť nástroj:
`./xtractor-cli.py -r [záznam] -c [prikaz]`

Filter

Nástroj ponúka tiež možnosť filtrovania. Filter je definovaný v parametri `-f` a má dve možné formulácie. Prvou pre získanie času je `"time dd-MM-yyyy-hh-mm dd-mm-yyyy-hh-mm"`, kde hh reprezentuje hodiny od 00 do 23, mm reprezentuje minúty od 00 do 59, dd reprezentuje deň v mesiaci, MM reprezentuje mesiac od 01 do 12 a yyyy reprezentuje rok. Tento filter vyberá prvky, ktoré majú svoj časový atribút v rozmedzí medzi týmito dvomi dátumami.

Tento filter sa viaže na nasledujúce atribúty:

- Záznam zariadenia `device`, kde filtruje na základe času extrakcie pri použití príkazu `list-device-records`.
- Záznam aplikácie, kde filtruje na základe času posledného popužitia pri použití príkazu `find-applications`.
- Záznam súboru, kde filtruje na základe času poslednej modifikácie pri použití príkazu `find-files`, `pull`.

Druhým možným filtrom je podľa názvu vo formáte `"name [nazov]"`. Parameter `[nazov]` reprezentuje časť názvu daného záznamu. Tento filter sa viaže s rovnkým použitím ako predchádzajúci filter.

Príklad:

```
./xtractor-cli.py -d [deviceID] -c find-applications -f
    "time 09-10-2017-00-00 10-10-2017-00-00"
```

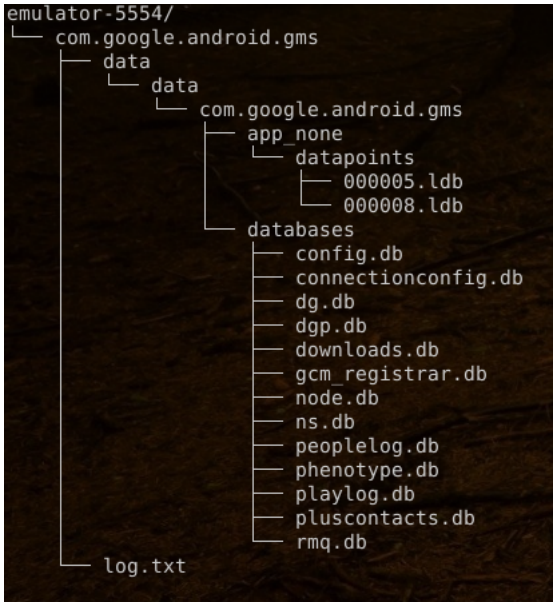
Tento príklad načíta záznamy všetkých aplikácií do databázy a vypíše všetky aplikácie, ktoré boli naposledy použité medzi 9.10.2017 a 10.10.2017.

3.3 Výstup

Výstupná zložka má nasledovnú štruktúru:

```
<výstupná_zložka>/
    cache.db
    <identifikátor_zariadenia>/
        <identifikátor_aplikácie>/
             <log.txt>
             <celá_cesta_súboru>/
```

Príklad reálneho výstupu, ktorý je zobrazený pomocou nástroja `tree`:



Obr. 3.4: Výstup

Jednotlivé logy sú odelené reťazcom `"\n----------\n"` a sú uložené v nasledovnom poradí:

1. package
2. logcat
3. batterystats
4. activity
5. procstats
6. appops
7. notification

Formát logu:

```
-----------
appops:
    Package com.android.chrome:
      COARSE_LOCATION: mode=0
      FINE_LOCATION: mode=0; time=+2h54m22s217ms ago
      VIBRATE: mode=0; time=+4d2h12m50s861ms ago; duration=+46ms
      POST_NOTIFICATION: mode=0; time=+2h54m53s333ms ago
      READ_CLIPBOARD: mode=0; time=+4d2h12m49s834ms ago
      WRITE_CLIPBOARD: mode=0; time=+4d2h12m56s433ms ago
      TAKE_AUDIO_FOCUS: mode=0; time=+2h54m53s357ms ago
      WAKE_LOCK: mode=0; time=+2h54m15s951ms ago; duration=+14ms
      MONITOR_LOCATION: mode=0; time=+2h54m21s866ms ago; duration=+366ms
      TOAST_WINDOW: mode=0; time=+2h56m37s986ms ago; duration=+4s4ms

-----------
notification:

-----------
```

Obr. 3.5: Formát logu

Kapitola 4

Moduly

Nástroj sa skladá z niekoľkých modulov. Táto kapitola dopodrobna rozoberie ich fungovanie.

4.1 ADB

Tento modul je založený na nástroji adb, ktorý musí byť nainštalovaný na zariadení. Slúži na zadávanie adb príkazov, ktoré sa vykonajú na skôr zvolenom zariadení. Tento modul umožňuje zistiť, či je zariadenie rootnuté, či sa na zariadení nachádza určitý príkaz. Hlavným cieľom modulu je však vykonávať adb príkazy na zariadení. Toto poskytuje metóda `query`, ktorá musí niektoré príkaz mierne pozmeniť, aby ich bolo možné vykonať. Problém tvoria zariadenia, ktoré sú produkčnými zostavami (Production build). Pre tieto zariadenia nie je možné získať root cez príkaz `adb root`. Pre tieto zariadenia je preto potrebné `shell` príkazy spúšťať cez `adb shell su -c <prikaz>`, kedy sa najprv priamo na zariadení spustí root a až potom sa sputí príkaz. Root je totiž nevyhnutný ak sa pristupuje ku systémovým dátam.

Ďalším problémom, ktorý kvôli tomuto vzniká je sťahovanie systémových dát cez `adb pull`, ktorý nie je možné vykonať ak `adb` nie je spustené v root režime. Na zriadení sa preto vytvorí dočasná zložka `/sdcard/tmp/`. Do tejto zložky sa premiestnia prenášané súbory, ktoré sú následne extrahované pomocou klasického `adb pull`, keďže z tejto zložky je možné sťahovať súbory aj bez root práv.

Jedným z problémov bol aj znak koncu riadka, ktorý sa mení v závislosti na operačnom systéme. Pri práci na tomto nástroji boli objavené 3 rôzne variácie. Sú nimi `"\n"`,`"\r\n"` a `"\r\r\n"`. Táto pohľadávka je uspokojená v metóde `checkEOL()`.

4.2 Device

Modul `Device` predstavuje samotné zariadenie, ktoré sa skladá z dvoch tried `Device` a `App`. Ako už bolo spomínané v odstavci 3.1.2, v prvej fáze sa automaticky stiahnu potrebné dáta. Objekt triedy `Device` obsahuje informácie o zariadení a objekt triedy `App` obsahuje informácie o aplikácii.
O každom zariadení `Device` sa uloží:

- `apps` - zoznam aplikácií, každá je tvorená objektom triedy `App`. Jeho obsah je popísaný ďalej.
- `version` - verzia systému Android
- `identifier` - identifikátor zariadenia
- `dirs` - zoznam zložiek pre dáta jednotlivých aplikácií
- `root` - boolean existencie rootu
- `EOL` - znak alebo séria znakov, ktorá ukončuje riadok

O každej aplikácii `App` sa uloží:

- `ID` - plný identifikátor
- `lastUsed` - posledné použitie aplikácie
- `name` - názov aplikácie, získané cez aapt
- `dirArr` - zoznam zložiek využívaných danou aplikáciou
- `dataFiles` - zoznam súborov danej aplikácie, štrukturovaný je ako slovník:
  - `fullPath` - celú cestu ku súboru
  - `name` - meno súboru
  - `size` - veľkosť súboru
  - `time` - čas poslednej úpravy súboru
- `activityName` - názov aktivity spúšťanej aplikáciou
- `log` - slovník obsahujúci logy
- `UID` - UID, ktoré patrí aplikácii
- `PID` - zoznam PID, ktoré patria aplikácii

4.2.1 Device

Trieda `device` predstavuje samotné zariadenie a vykonáva všetky potrebné úkony.

Hľadanie názvov

Pri každej aplikácii sa ukladá plný identifikátor, dátum a čas posledného použitia. Taktiež sa pomocou nástroja `aapt` pokúsi na zariadení získať plný názov aplikácie. Tento nástroj sa však nemusí na zariadení nachádzať. Ak sa na zariadení nenachádza je na zariadenie vložený z lokálnej zložky `aapt/`, kde sa nachádzajú binárne súbory nástroja. V prípade, že tento postup nie je možný, je nástroj možné použitý lokálne na počítači. Pre využitie tejto aplikácie priamo na počítači je potrebné mať prístup ku `.apk` inštalačnému súboru. Sťahovanie `.apk` súborov zvyšuje časovú náročnosť tohto procesu. Pokiaľ už názov aplikácie bol zisťovaný v niektorom predchádzajúcom použití nástroja, použije sa tento údaj z lokálnej databázy `cache.db`.

Hľadanie zložiek

Nástroj hľadá zložky, ktoré môžu byť využívané jednotlivými aplikáciami. Toto hľadanie sa riadi konvenciami Android aplikácii. Zložky sú postupne hľadané v zložkách `/data/data/`,`/sdcard/Android/data/`,`/sdcard/Android/media/`,`/sdcard/Android/obb/`,`/sdcard/`. V týchto sa hľadá plný identifikátor. Špeciálnym prípadom je posledná spomínaná zložka `/sdcard/`. V tejto zložke majú vývojari voľnosť nad zložkami a ich názvami. Z toho dôvodu sú tu vyhľadávané zložky, ktoré majú buď plnú alebo aspoň častočnú zhodu s verziou názvu aplikácie alebo časťou identifikátoru, ktorá nie je kľúčovym slovo. Takými slovami sú napríklad ändroid","google", pretože tieto slová sa vyskytujú vo viacerých identifikátoroch, a teda nie sú jedinečné. Nad touto množinou potenciálnych názvov je ešte treba zapracovať, lebo názov zložky nemusí mať nič spoločné s názvom aplikácie, takže pokrytie ešte nie je stoprecentné.

Hľadanie súborov

V nasledujúcom kroku sa tieto zložky rekurzívne prehľadajú pomocou nástroja ls a záznami o všetkých súboroch sú uložené aj s názvom, plnou cestou, veľkosťou, dátumom a časom poslednej modifikácie. Pomocou týchto informácií je potom možné súbory následne filtrovať. Súbory sú momentálne dostupné iba v rámci jednotlivých aplikácií.

Hľadanie logov

Nástroj taktiež vyhľadá logy pre jednotlivé aplikácie. Log sú získané pomocou dvoch nástrojov. Sú nimi `logcat` a `dumpsys`. Logcat obsahuje log, kedy aktivít aplikácií. Dumpsys dokáže vypísať log rôznych služieb na zariadení. Pre ich informačnú hodnotu som vybral nasledovné.
Služby cez dumpsys:

- `package` - imfomácie o aplikácii ako balíku, jej aktivity, poskytnuté Content Providers (Poskytovatelia obsahu), práva. Obsahuje tiež čas inštalácie a aktualizácie aplikácie. Okrem iného obsahuje cestu ku `.apk` súboru. Keďže sa formát menil od verzie 5.0, jej spracovanie závisí na verzii systému Android.
- `appops` - informácie o tom, kedy a na ako dlho aplikácia naposledy vykonala svoju určitú aktivitu.
- `batterystats` - informácie o spotrebe danou aplikáciou. Obsahuje spotrebu dát, procesorového času, ako dlho bola spustená ako aj služby, ktoré využila.
- `activity` - aktivity spojené s danou aplikáciou
- `procstats` - informácie o spotrebe procesoru danou aplikáciou
- `notification` - informácie o momentálne zapnutých notifikáciách na zariadení. Z jej logu je možné vyčítať čas a obsah danej notifikácie.

4.2.2 App

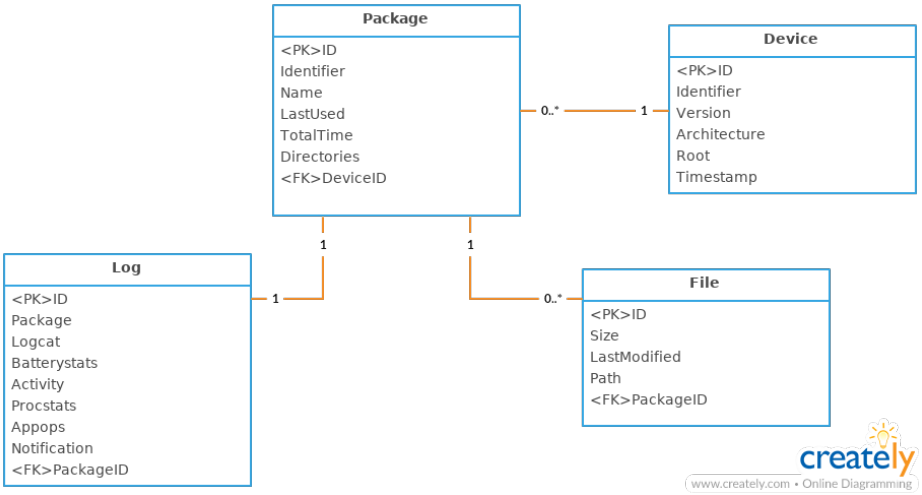
Trieda slúži na uschovanie dát jednotlivých aplikácií. Keďže sama nevykonáva adb príkazy, takže nie je závislá na triede adb. Okrem ich uschovania a vypísania dokáže aj zoradiť súbory podľa názvu, veľkosti a dátumu modifikácie.

4.3 Cache

Tento modul slúži pre prácu s lokálnou **sqlite** databázou `cache.db`. Táto databáza slúži na ukladanie dát z extrakcie dát z mobilného zariadenia. Každá extrakcia vytvára samotný záznam s časovou známkou. Databáza obasahuje 4 rôzne tabulky. Tabuľka `Device` obsahuje záznamy o extrakcii, pri každej extrakcii dát je zariadenie brané ako jedinečné. Tabuľka `Package` obsahuje informácie o aplikácii. Tabuľka `Log` obsahuje záznamy logov je pre jednotlivé aplikácie. Tabuľka `File` obsahuje informácie o súboroch spätých s aplikáciou. S databázou sa komunikuje pomocou sady setterov a getterov.

Okrem samotného ukladania dát sa tento modul stará aj o filtrovanie dát na výstupe. Filter musí byť v takom prípade nastavený ešte pred samotnou extrakciou. Filtrovanie na úrovni je využitý z toho dôvodu, že časovo menej náročný oproti filtrovaniu na bázy samotného programovacieho jazyka.

4.3.1 Schéma



Obr. 4.1: Schéma databázy

Kapitola 5

Potrebné dokončiť

- Ďalšie zložky ku každej aplikácii
- Root na zariadení