

こわくない Auto Backup

Tips of Auto Backup

コンテンツ

自己紹介

Auto Backup の概要

Auto Backup こわい

とにかくこわい人向け

こわくない Auto Backup

自己紹介

自己紹介

酒多 優太 (22)

Retty で Android に関係することとかやってる大学生

Watch OS 2 向けの Retty の開発をスクラッチでやった



Auto Backup って？

Auto Backup の概要

- 新機能としてMarshmallow (API レベル 23) から実装された
- Target SDK Version が Marshmallow 以降だと動く
- ややこしい Backup API を使わなくてもよくなる
- コードの変更なしに、自動でアプリのデータをクラウドにバックアップしてくれる
- Backup は Google Drive に暗号化されて置かれる
- App につき 25 MB まで利用可能で超えると Backup されなくなる
 - ◆ ユーザーの Google Drive 容量は消費されない

Backup されるタイミング

前回の Backup から 24 時間経っていて

Device がアイドル状態で

充電中で

WiFi ネットワークに接続している時

Restore されるタイミング

新しい端末に App をインストールした時

App を再インストールした時

Backup されるもの

デフォルトでは

キャッシュ `Context#getCacheDir()` `Context#getCodeCacheDir()`

外部ストレージのデータ `Context#getExternalFilesDir()`

を除く App に作られるデータすべて

Restore されるもの

Google Drive にある最新の Snapshot

Auto Backup こわい

ここがこわいよ Auto Backup

- デフォルトだとすべきでないものも Backup される
 - ◆ 端末固有の情報が別の端末に Restore される可能性
 - ◆ Session 情報など期限付きなもの
- DB など構造が変わり得るものの Backup あぶない
 - ◆ No such table
- 25 MB 超えると Backup されないって書いてるけど具体的な挙動が全然わからない
- ユーザが On/Off できるのである前提での設計はできない

起こりうる問題

例えば.....

Push 通知が上手く動かない

Backup が全然仕事してなくて App をふと再インストールしたらかなり昔のデータに
ロールバックからの原因がわかりづらいクラッシュや不具合

こわい！！！！

とにかくこわい人向け

```
<manifest ...>  
  ...  
  <application ...  
    android:allowBackup="false">  
      ...  
    </application>  
  ...  
</manifest>
```

こわくない Auto Backup

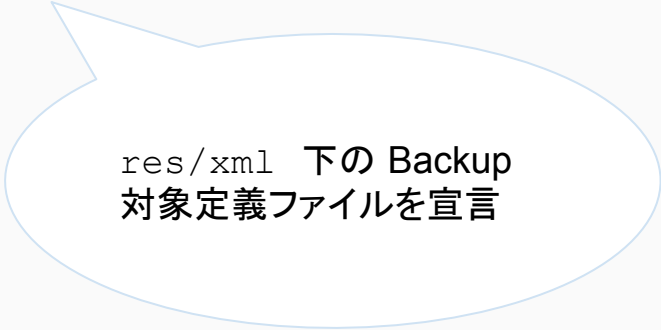
この辺がわかればこわくなさそう

- Backup すべきでないもの
- Auto Backup への認識
- 25 MB 制限を超えた時の挙動

Backup すべきでないもの

- 端末固有の情報
 - ◆ 別の端末で Restore された時に不整合が生じる
- リアルタイム性があったり、順番が大事だったりするフラグ
 - ◆ 同期的でないので思わぬ不具合の原因になる可能性がある
- メディアファイルなど大容量のもの
 - ◆ 25 MB しか使えないのでそもそも向いていない
- バージョンに依存する構造を持つもの

```
<manifest ...>
  ...
  <application ...
    android:fullBackupContent="@xml/backup">
      ...
    </application>
  ...
</manifest>
```



res/xml 下の Backup
対象定義ファイルを宣言

```
<?xml version="1.0" encoding="utf-8"?>
<full-backup-content>
  <include domain="file"/>
  <exclude domain="file" path="hoge"/>
</full-backup-content>
```

大雑把に言うとホワイトリスト
これで指定したものが Backup されるようになる

include の逆
これで指定したものは Backup されなくなる
同じターゲットに include と exclude を指定した場合
exclude が優先されるのでこのケースだと file 配
下の hoge を除くファイルが Backup される

Backup 対象ファイルの種類を表す
以下の種類がある

```
root
  App のルートディレクトリ
file
  Context#getFilesDir()
database
  Context#getDatabaseDir()
external

Context#getExternalFilesDir()
sharedpref
  SharedPreferences
```

Optional
指定すると domain の下の特定のファイルを
対象にできる

Auto Backup への認識

- クラウドと同期するためのものではない
 - ◆ あくまでも Backup
- インストールし直すたびに同じ設定するのめんどくさそうだし対応しようくらいの気持ちでいるのが良い
- あったら嬉しいけど別になくてもいい
- 絶対に Backup されなければいけないものや同期を取らなければいけないものには別の手段を用意すべき

25 MB 超えた時の挙動

公式ドキュメントには

“Each app can store up to 25MB. Once its backed-up data reaches 25MB, the app no longer sends data to the cloud.”

とかいてある

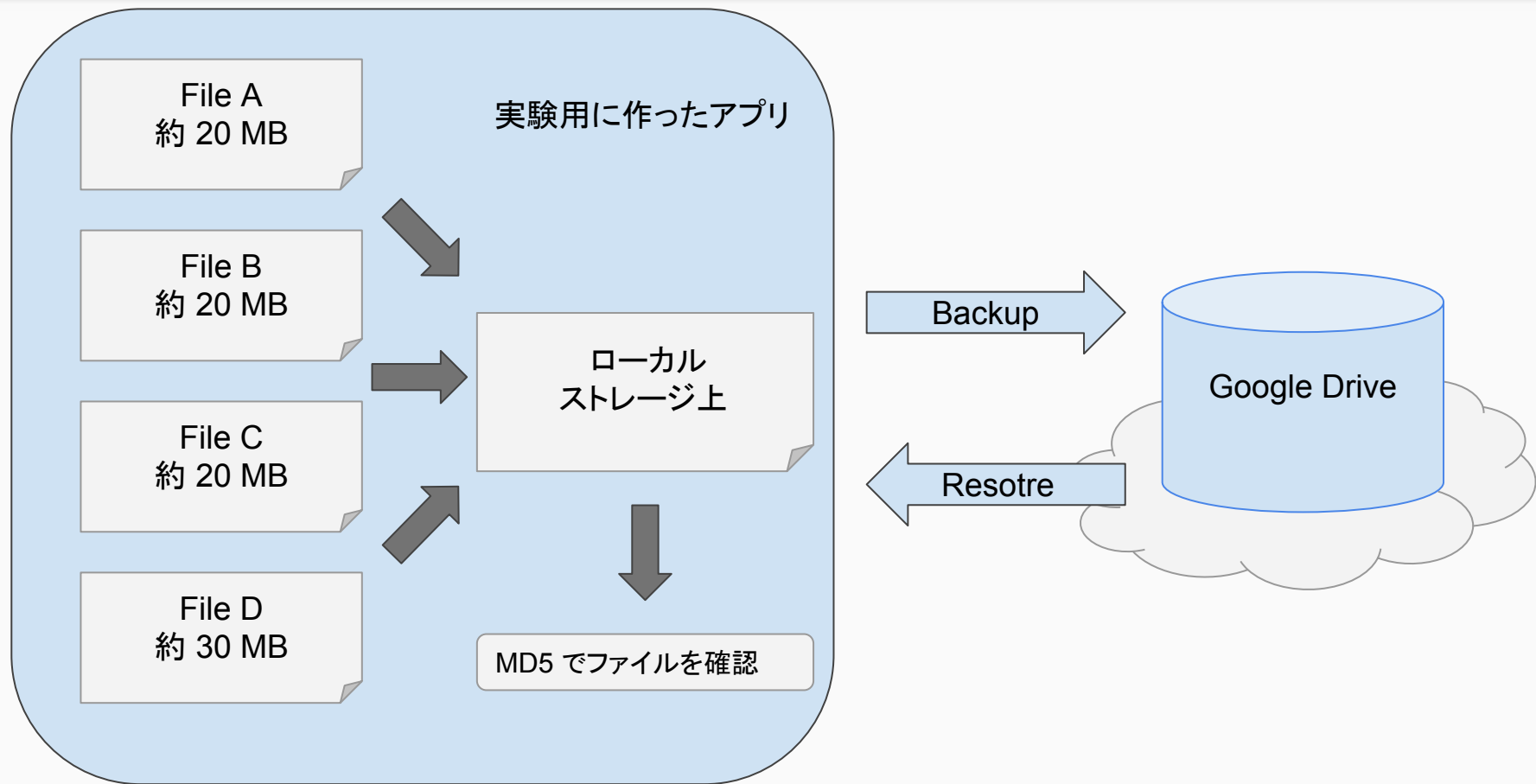
なにはななくともソースを読んでみる

`com.android.server.backup.BackupManagerService` が
`com.android.internal.backup.IBackupTransport` を実装したクラスの
インスタンスのメソッドを呼ぶことで実現している

自分の環境 (Nexus 6 Android 6.0) のAuto Backup では
`com.google.android.gms.backup.BackupTransportService`

肝心のソースが読めない！！

ということで実験してみる



用意したもの

→ 適当なダミーファイルを 3 つ

- ◆ 今回は `head -c $SIZE /dev/urandom >> filename` で作成

→ 実験用アプリ

- ◆ Assets から用意したファイルをデータとして端末に書き込む機能
- ◆ 書き込んだファイルを Auto Backup の対象に設定
- ◆ MD5 を計算する機能

→ 強制 Backup するコマンドの情報

- ◆ `adb shell bmgr`

実験 1

1. 実験用 App をインストールした
2. `adb shell bmgr run` を実行した
3. File A (20 MB) をローカルにコピーして
`adb shell bmgr fullbackup $PACKAGE_NAME`
で Backup を実行した
4. File B, C (各 20 MB) について File A と同じ path に上書きし、同様の操作をした
5. App のデータを削除したあと Backup からファイルを復元して MD5 ハッシュを計算し、どのファイルが復元されたか確認した

実験 1 結果

復元されたのは File C で、直感通りの結果になった

実験 2

1. 実験 1 の 1 ~ 5 までを行った
ただし、File C の代わりに File D (30 MB) を用いた
2. File A についてもう一度操作を行い、Restore して MD5 ハッシュを確認した

実験 2 結果

どちらの Restore されたファイルもハッシュ値は File B と同じものだった

File D の Backup 以降何度 Backup を試みても logcat に

```
PFTBT      : Transport rejected backup of PACKAGE_NAME
```

と表示されるようになり、Backup が上書きされることはなくなった

Local

Cloud

File A (under 25 MB)



File A

File B (under 25 MB)



File B

File D (over 25 MB)



File B

Any File



File B

考察

一度でも Backup 対象のデータが 25 MB を超えた状態で Backup が実行されると、対象データの容量が 25 MB を下回っても Backup できなくなる

(そのままの意味だった)

sharedpref 以外の大きくなりうる domain を対象に include する場合は path を指定しておくと事故が起こらず無難そう

まとめ

こうするとこわくない

大きいデータを扱わない App の場合は Backup するのがふさわしくないデータを exclude するだけでOK

そうでない場合は Backup するのがふさわしいファイルを path まで指定して include してあげると不意な事故を防げる

ちょっとだけ UX を改善してくれるだけのものと割り切ろう

おまけ

デバッグ中に Backup できなくなって公式ドキュメントの方法で復帰できなかった場合

設定 -> バックアップとリセット -> データのバックアップ を off/on

して Google Play 開発者サービスのデータを削除して再起動すると再びバックアップできるようになる (ことがある)

Appendix

実験で使った App

https://github.com/YutaSakata/android_m_autobackup_lab

おわり