

File Synchronization using API Google Drive on Android Operating System

Agustinus Noertjahyana, Kevin Darmawan, Justinus Andjarwirawan
Informatics Engineering Department
Petra Christian University
Surabaya, Indonesia
agust@petra.ac.id

Abstract— Nowadays, most people use mobile devices to perform tasks and work. There are so many sites offering online storage. Google drive deliver online services with a large capacity today. Google Drive have lots of features that make them really competitive among all their competitors, that feature is the make user may edit edit the document on their site and share documents for collaborative work with others.

Google Drive online storage is created to answer the current lifestyle of people all over the world. Most people use more than one device. By using online storage, the data could be accessed online anytime anywhere. Problem that happens is people want to get the latest version of each file on each device. To overcome these problems could be use a file synchronization feature.

This research will focus on two way file synchronization. By using date comparison method, file with the latest modification date will be altering the older file. By implementation of this method both storage will always have the latest updated file. All synchronization and other processes such as compare, upload, download, share, and update will be done based on Google Drive APIs.

Keywords— *file synchronization, Google Drive*

I. INTRODUCTION

Many Android users enabling them gadgets like a computer, it is because the Android OS successfully brought the concept of "PC Experience" through the user interface. Therefore, many users use the Android smartphone to perform various tasks commonly performed on a PC. For example, read email, reply to email, document and photo-editing.

Problems occur when trying to copy files to the PC, the transfer method most often used by Android users is to use a USB cable or Bluetooth. If users want to copy data on multiple devices will certainly take a long time if copying the data one by one. Another problem is security, as many people know that the storage devices such as flash memory or micro SD are very vulnerable to the corruption of data.

To overcome these problems, many people use online storage to store their data. As one of the popular online storage, Google drive provides services free of charge. With Google Drive, users will be facilitated when transferring documents to the PC and vice versa. Users can also share documents without

using email attachments. Documents or files that are stored on Google Drive can be easily moved to various devices. Permissions of the user can be limited to only view or edit.

More and more users are utilizing google drive for file synchronization, file types are also more diverse synchronized either type. Doc., Xls., Ppt and. Pdf.

Google Drive has a native client for many platforms including Windows PC, Mac, and Android. But for the OS that do not have native client can access the web version of Google Drive. Unfortunately the synchronization feature that can allow a user to transfer files between local storage and Google Drive, only on Windows and Mac client.

As for the Android client is only able to upload single files as well as the make / set offline documents. The ability to upload a single file is wasting time if many documents that are uploaded. Similarly, the make process should also be done offline one by one file.

II. THEORY

A. Android Application Fundamental

Android is a "stack" of mobile software device that consists of an operating system, middleware, and application support [1]. In recent years increasingly well-known name in the Android community a variety of the world, including Indonesia.

Android architecture as a stack with the application is at the top and the Linux Kernel on the bottom, as can be seen in Figure. 1.

Developers can access the framework API which also used by the core applications. Application architecture is designed to facilitate the use of other application components.

In Android there is a set of core libraries that provides functionality very similar to those of the core Java libraries. Every Android application runs in its own process, and at the instance of the Dalvik Virtual Machine [2]. Dalvik was created so that a device can run multiple VMs efficiently. Dalvik VM executes files in the DEX (Dalvik Executable), which is optimized for minimum memory usage.

Dalvik VM relies on the Linux kernel to run some functionality such as threading and low-level memory

management. Although the program is running in a virtual machine DEX format, however the Android installation file has a different format, APK (Android Package), and each file generally represent a single APK application. APK file will be compiled through the Android SDK tools, along with the data and resources.



Figure 1. Application Stack – Android Architecture

B. Android Application Lifetime

Once installed in the device, each Android application runs in a security sandbox:

- Android OS is a multi-user, where each application has a different user.
- In general, the system gives each application a unique user ID. System set permissions for all files in the application can only be accessed by the user ID applications.
- Each process runs in its own virtual machine. So that an application code runs isolated from other applications.
- Each application runs its own Linux root process. Android will execute the required application components and will shut down applications that are not running or the system needs more memory.
- An application can access the data device such as contacts, SMS messages, SD card, camera, Bluetooth. All permissions for access must be granted by the user when performing install.

C. OAuth 2.0 Protocol to Access Google APIs

In order to access online services, user must be authenticated to the service, required proof of the truth of identity. For applications that access the third-party service, security issues become more complex.

Not only on the user must authenticate to access the service, but the application must also be given permission to access by

the user. Standard industry to commit to service third party authentication is OAuth protocol [3]. OAuth provides a value, authentication token, which represents the user's identity and access rights applications. OAuth is best used for : Obtain permission from the user to access the service online using a user account, authenticate the user to the online service and handling error in authentication process.

Google API using OAuth 2.0 protocol to perform authentication and authorization. Google supports the OAuth 2.0 flow, which includes a web server scenario, java script, device, installed application and server to server. OAuth 2.0 is a simple protocol and developers can integrate the Google OAuth 2.0 endpoint easily.

D. Google Drive API

Google Drive API is a library that have the function to help developers of doing things in Google Drive, such as creating a new file, download, upload [4]. One of the most important parts of Google Drive is a file storage, therefore most of the Google Drive have the function to get data from a file.

Although the API to get the data from the file, but the API does not directly access the files. API just get the metadata of a file or folder. This metadata is used to obtain data on file. Data types used in the file is not derived from java IO, but the types of data files from SDK.

The methods used to handle the resource file are:

- Get: Used to get the file by using the file ID. Get can also be used to download files, by calling File.GetDownloadUrl () and parse the data in an HTTP Response variable.
- Insert: Insert function, can be used to upload files to Google Drive. There are several parameters that are included such as OCR to set whether the files are sent to be processed by Optical Character Convert Recognition or when the file is sent to be modified into a Google Doc format. Title and description files should be set before sending the file.
- Patch: Used to update the metadata of the file.
- Update: It has the same function with the patch, but the update can be used to change the content of the file.
- Copy: Used to create a copy of a file. API callings can be done by copying files().copy (original FileID, destination files).
- Delete: Used to delete a file by file ID. Delete Files are not put in the trash, causing the risk of losing files.
- List: Used to download any file list contained in Google Drive. If there is no query then by default it will read all the files and folders in Google Drive including those already in-trash.
- Touch: Used to update the time based on server time.
- Trash: Move the file to the trash. There is also the opposite of Untrash to restore files from trash.

Google Drive API can also be used to obtain other resources such as:

- **About:** This Resource contains user data. Information such as the name and the remaining quota.
- **Change:** a resource that records all changes or file updates. There are two methods to get the Change, by using the Get or List.
- **Children:** A resource used in recording the contents of a folder or a recording of the contents of the file. There are four methods that can be in implementations such as Get, List, Insert, Delete. Insert is a good option, if the file was uploaded just put into one folder only (one parent).
- **Parent:** Resource to record the parent or folder. There are 4 methods: Get, List, Insert, and Delete. Insert has a function to create a folder and put the files into it.
- **Permission:** Resource which function to record the permissions of a file. Each permission has its own ID, and can be modified by the method of update or insert.
- **Revision:** Resource that records any revisions or updates to the file. All revisions can be obtained by methods LIST, UPDATE, INSERT and DELETE.

III. SYSTEMS DESIGN AND IMPLEMENTATION

System using the activity, service and receiver component. Activity is a component system that focuses on the user interface, service is a component that runs in the background, and the receiver is a component that helps service. Components in this system are:

- Activity Main Drivenote.
- Activity List Local Storage.
- Activity List Drive File.
- Activity Note Editor.
- Activity Share Files / Folders.
- Background File Service Synchronization.

Minimum requirement of this application is Android OS smartphone that has a minimum of version 2.2. Application requires internet only when opening drives Activity List File. Application can still run in offline mode. Users can edit the text by using the text editor for the local file, or use the Google Docs editor for files located on Google Drive.

A. Main Activity Drivenote

Main Activity Drivenote display when the application is first run.. This interface has a very simple design with 4 buttons for:

- Open the Activity List Local Storage.
- Open the Activity List File Drive.
- Open the Note Editor.

- View a list of files that are synchronized.

Users can set a synchronization schedule or triggering synchronization manually. Setting process and trigger synchronization is performed using the android menu. On the menu there is an option to synchronize based on certain hours each day, certain time intervals, and the last trigger manually. Users can make changes to the account that will be used for login Google Drive. Main Activity Drivenote can be seen in Figure 2.

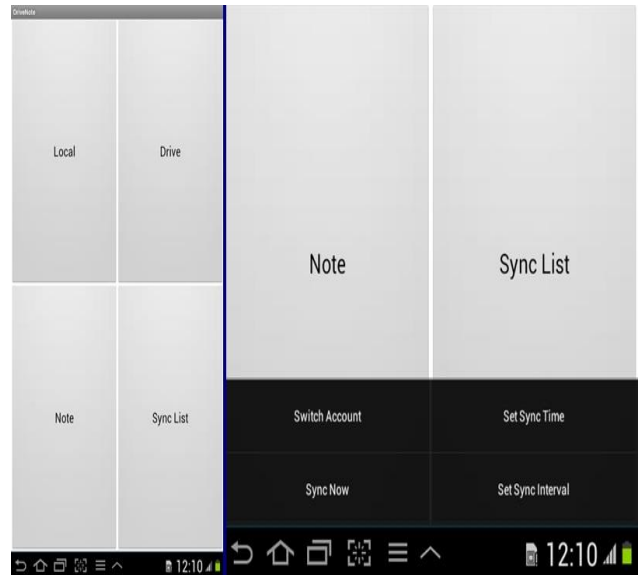


Figure 2. Activity Menu Drivenote Interface.

B. Activity List Local Storage

Activity List Local Storage on this application has a function to explore the local storage of device and file management. An important function of this activity is to record the local file which will be in sync with Google Drive. Recorded files will be stored in a SQLite database that is available on any Android OS.

The following are functions of this activity: Conducting Listing on Local Storage, Perform Navigation in Local Storage, Open the file, according to the file types that can be recognized by the device, View the properties of a file, File Management, include the Create Folder, Copy, Cut, Paste, Delete, and Rename, Change the Display Order, Sort by Name, or date modified file and Listing files to be synchronized.

To perform the local file listings can be done by calling the function `File.listFiles()`, similar to that of the Java language. This function will display a list of files from a folder. The file list will be processed to determine the appropriate icon for each file and check for checkbox icon, the whole process is covered in class display adapter. So, display adapters featuring an interface to the user. At the interface can also be found checkbox in each file, this checkbox serves to record the files to be manipulated or synchronized. Copy, cut, paste, rename, and delete files can also be done using the menu contained this activity. File listing mechanism can be seen in Figure 3.

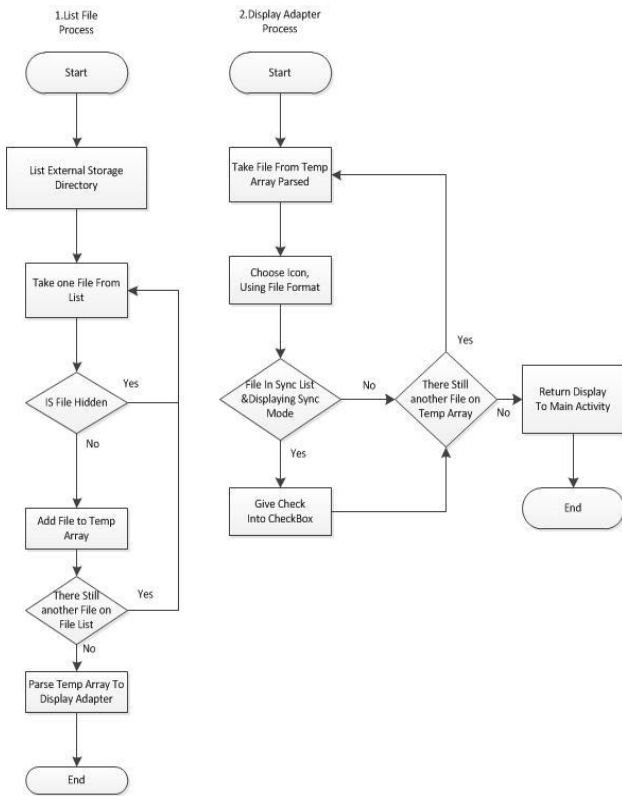


Figure 3. File Listing Mechanism

C. Activity List Drive File

Activity List File Drive has a function to display the contents of a user on google drive and has a function as Google Drive File Explorer. Users can navigate the file, similar to the activity list of local storage. However, there are reductions to management functions, which can only rename the file or folder. At this activity has the function to save the files to be synchronized to the local memory. The result will be recorded on the database SQLite.

Here are the functions that are provided in the file drive activity list: List the content of the user who is logged on, Navigation in Google Drive, To share files or folders with other users, Open the file, according to the file types that can be recognized by the device, Rename and trash files / folders, and Perform download files.

Steps to list file is as follows:

- Registering the application that is built into the Google APIs Console. After the registration process will receive an API Key, which is used to access the API drive. API Key is used by Google to identify the applications that access the API.
- Require authentication User access to Google Account. Once authenticate then the application will have permission to access the content of Google Drive account. Once the authentication process is completed, will receive an OAuth token. This token is used to identify the user API access.

- Perform drive service, which is an object that is created from a class drives used to handle queries in Google Drive.
- Perform Query API to drive service. Almost all queries will generate metadata file only.

Activity List Drive Files Flowchart can be seen in Figure 4.

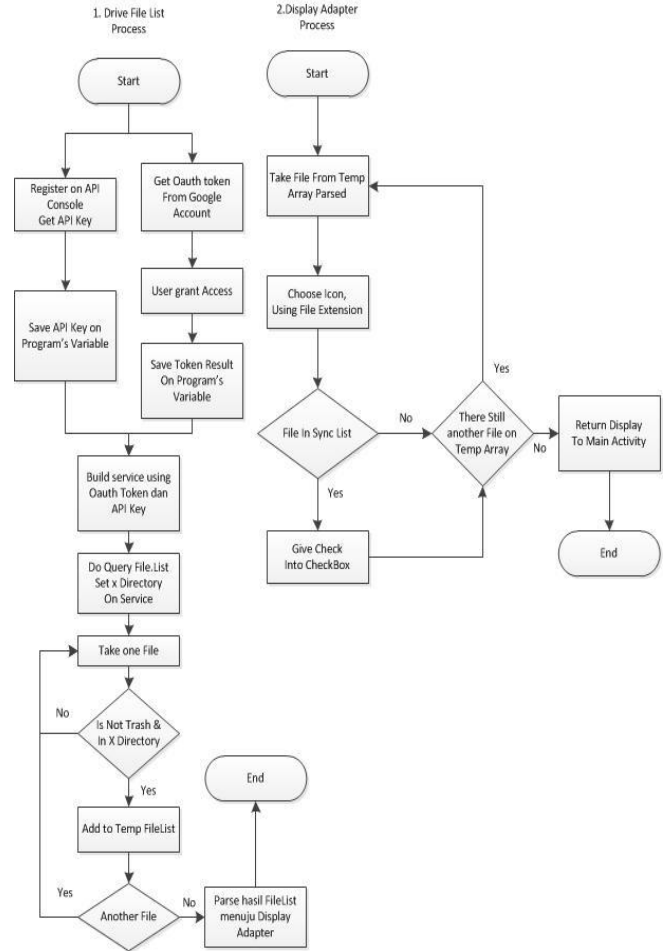


Figure 4. Activity List Drive File Flowchart

D. Activity Note Editor

Note Activity Editor has functions to create and edit a document file types. User can write and then save the notes into a text file with the file format. "txt". The application will save the new file in the root folder, while for existing files will be stored in the original folder.

In the storage process, a possibility is as follows:

- If the title and the content is not empty, and the file is not in sync database, then the file will be created and the application will ask if the user wants to synchronize with Google Drive.
- If there is a title, but no content, the text file will be created with empty content.

- If there is no title, then the file will not be created, and the remind message will appear to give the name of the file.

E. Activity Shares Files / Folders

Activity share files / folders are used to share the file or folder to a specific user. This activity can be activated by long press on a list item in the File Drive.

In this activity will display users who have access rights to the files in a list. Users can add a particular user in this activity to share files. Google Drive divides the right of access to the file as follows:

- **Owner:** These permissions indicate that the user is the owner of the file, changing the user's privileges could not be done. Users with owner privilege is able to modify the file and also open / read the file.
- **Writer:** These permissions indicate that the user has full access to the file so that the user can modify the file.
- **Reader:** These permissions indicate that the user can open the file but can not modify or alter the file.

F. Background File Service Synchronization

Background File Synchronization Service is running a background service that has been running when the requirements are met. Background service that is run is vital for running the file synchronization process. This component requires an internet connection because they have to check for the latest files, the new file will replace the older file. It should be noted that the Google Drive, a longer file will pinned with newer files due to API calls Update ().

The background service will read the contents of SQLite database to determine which files should be in sync. The file will continue to be updated in the latest synchronization, while the old one will be removed pinned with newer files either on Google Drive or on the device. Service will also distinguish which files should be in sync based on the account that is currently logged on.

To facilitate a comparison between files then each file within the entity has a ListID, ListId indicator is used for synchronization. ListID has the same value on the two entities for the same file. Simply by taking one file, and then compare it with other files that have the same ID on the entity.

Synchronization process can be seen in figure 5.

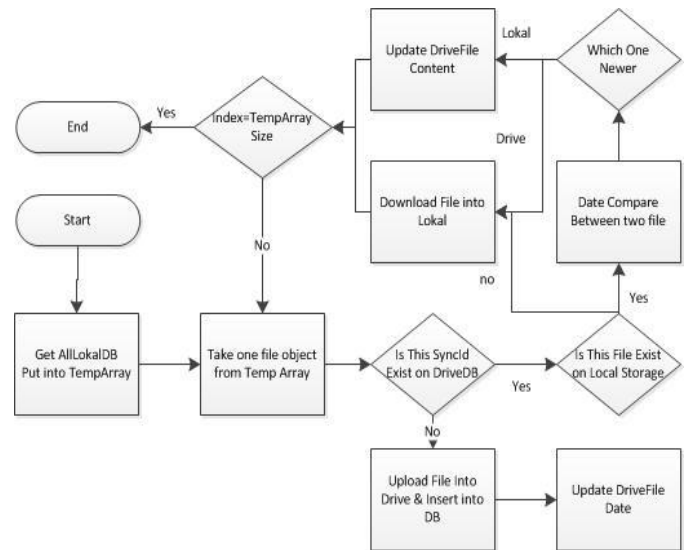


Figure 5. Synchronization Process

IV. CONCLUSION

Based on the results of tests performed in the application can be summarized as follows:

- The time required to access Google Drive could be longer if the token is expired, because the application will restart and request a new token.
- Files that are downloaded through the synchronization process will automatically go into the sync list. For the downloaded files manually via the context menu is not included in the synchronization list.
- Because Google Drive is able to perform auto-touch on the file, then the file is located on Google Drive may change the date of last modification though none modify it.
- If the connection is lost when synchronization is running, the application will not be an error but the file is currently being downloaded process will be damaged and could not be opened.

REFERENCES

- [1] Android Developer. <http://developer.android.com/about/index.html> Accessed on December, 2012.
- [2] Y. Ashish et al, "Dalvix - Virtual Machine", Indian Journal of Engineering, 2012, 1(1), pp100-104.
- [3] Google Developers, "Using Oauth 2.0 to Access Google APIs", <https://developers.google.com/accounts/docs/OAuth2>, 2011.
- [4] Google Developers, "API Reference", <https://developers.google.com/drive/v2/reference/>, 2012.