

ITPlusPoint

Android Sync

Sync Methods on android

Sanjeet



14

1.Sync Options on Android:

1. Using the Sync Adapter frame work
2. Using the AsyncTask and Intent Services

1.1Sync Adapter Framework:

Synchronizing data between an Android device and web servers can make your application significantly more useful and compelling for your users. For example, transferring data to a web server makes a useful backup, and transferring data from a server makes it available to the user even when the device is offline. In some cases, users may find it easier to enter and edit their data in a web interface and then have that data available on their device, or they may want to collect data over time and then upload it to a central storage area.

Although you can design your own system for doing data transfers in your app, you should consider using Android's sync adapter framework. This framework helps manage and automate data transfers, and coordinates synchronization operations across different apps. When you use this framework, you can take advantage of several features that aren't available to data transfer schemes you design yourself:

1.1.1.Pros:

Plug-in architecture

- Allows you to add data transfer code to the system in the form of callable components.

Automated execution

- Allows you to automate data transfer based on a variety of criteria, including data changes, elapsed time, or time of day. In addition, the system adds transfers that are unable to run to a queue, and runs them when possible.

Automated network checking

- The system only runs your data transfer when the device has network connectivity.

Improved battery performance

- Allows you to centralize all of your app's data transfer tasks in one place, so that they all run at the same time. Your data transfer is also scheduled in conjunction with data transfers from other apps. These factors reduce the number of times the system has to switch on the network, which reduces battery usage.

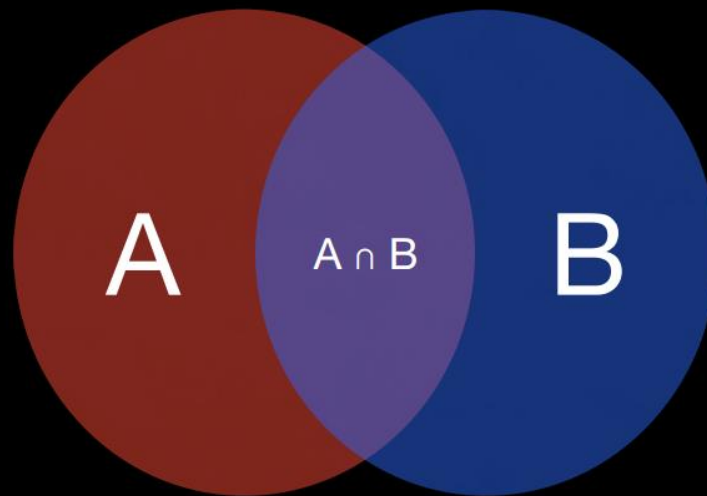
Account management and authentication

- If your app requires user credentials or server login, you can optionally integrate account management and authentication into your data transfer.

1.1.2.Cons:

- Sync adapters run **Asynchronously**.
- Transfer data regularly and efficiently, but **NOT** instantaneously.
- If you need to do real-time data transfer, you should do it in an **AsyncTask** or an **IntentService**.

2.Sync Methodolgy and Algorithms



Sync: In Depth

PREPARE

SYNC OBJECTS IN ONLY A

SYNC OBJECTS IN ONLY B

SYNC INTERSECTION

CLEAN UP

Needed for Sync

- On each device, each object needs:
 - Creation Date
 - Modification Date
 - UDID

PREPARE

- Establish Communication With Sources
- Grab summaries from A and B
 - UUIDs, creation, modification
- Sort into sets

SYNC OBJECTS IN ONLY A

- For each object **o** in **a**:
 - if **o.creation** > last sync then
 - tell **b** to copy **o** over
 - else
 - tell **a** to delete **o**
 - end if
- next

1. Set A is collection of all data related to the section,

For example-Set a will contain the all data of the workout and exercises.

SYNC OBJECTS IN ONLY B

- For each object **o** in **b**:
 - if **o.creation > last sync** then
 - tell **a** to copy **o** over
 - else
 - tell **b** to delete **o**
 - end if
- next

1. Set B is the collection of the data what we are getting from the server for section ,

For example-GetAllWorkoutsResult returns the collection of workouts and the corresponding exercises.

SYNC INTERSECTION

- For each object **o** in both **a** and **b**:
 - if **o.modification** < last sync then
 - skip it
 - else
 - if *only a's* mod > last sync then
 - propagate **a's** version to **b**
 - else if *only b's* mod > last sync then
 - propagate **b's** version to **a**
 - else if *both a and b's* mod > last sync then
 - present conflict
 - end
- next

1. By default last sync is "1970-01-01 00:00:00"
2. If set A has element having the modified date greater than the last sync, and the corresponding element on the web is having an older version than the last sync, It simply update the record on the web.
3. Similarly It as step 2 for the B.
4. In case of the elements on the Both set A and B having modified date newer than the last sync, It generates a conflict where we can decide which to keep as a synced one.
5. Deleted the items on the device if it is not existing on the server set(B), (I am doing this as per your suggestion).

CLEAN UP

- tell **a** and **b** we're finished
- store current time as **last sync**

1. Sets up the last sync time as the current time as once it finishes the sync.

Problems and concerns-

1. The record is not instantly being created or updated on the server. It goes on the web after the sync.
2. If I directly create /update a record on the server it is having a newer version than the last modified one and hence the sync again pulls or updates the record on the device which causes the delicacy.
3. Doing the operations, coping/updating the server version to the device and vice versa, is causing the lots of code complexity, basically for the exercise and workouts.
 - Updating /Creating a workout requires more number of calls to the server which is causing the overhead for the sync.
 - Similar in the exercise, It is having encoded image along with the data which causes the overheads.