

# SESSION 06

```
export PATH=$PATH:/Users/sajib/Library/Android/sdk/  
platform-tools/
```

## # CONTENT PROVIDERS

- Activities [ <https://developer.android.com/guide/components/activities/intro-activities> ]
- Services [ <https://developer.android.com/guide/components/services> ]
- Broadcast Receivers [ <https://developer.android.com/guide/components/broadcasts> ], [ <https://developer.android.com/reference/android/content/BroadcastReceiver> ]
- Content Providers [ <https://developer.android.com/guide/topics/providers/content-providers> ], [ <https://developer.android.com/guide/topics/providers/content-provider-basics> ]
  - Generate data on a user's phone and view it in an app built by you
  - A layer between an app and its datasource that easily provides access to its datasource.
  - Another level of abstraction that helps change the data source without much change to the way the data is accessed
  - Allow multiple apps to access, modify and use a single data source securely
- Using Pre-existing Content Providers
  - needs `<uses-permission android:name="">` in `AndroidManifest.xml`

- Steps for using a Content Provider
  - ◆ Get permission to use the content provider
  - ◆ Get the content resolver
    - ◆ Content Resolver [ <https://developer.android.com/reference/android/content/ContentResolver> ]
    - ◆ Difference between content provider and content resolver [ <https://stackoverflow.com/questions/18874801/what-is-difference-between-contentprovider-and-contentresolver-in-android> ]
    - ◆ Resolves which content provider to work with when your phone has multiple content provider
    - ◆ ContentResolver contentResolver = getContentResolver();
  - ◆ Pick one of four basic actions: CRUD
    - ◆ Cursor cursor = contentResolver.query(CONTENT\_URI, null, null, null, null);
      - ◆ <content\_uri>, <projection>, <selection>, <selection arguments>, <sort order>
        - ◆ projection: filters columns
        - ◆ selection: statement for how to filter rows
        - ◆ selection arguments: what to filter
        - ◆ sort order: determines the sorting order of the returned result
    - ◆ Read from the data: query()
    - ◆ Add a row or rows to the data: insert()
    - ◆ Update the data: update()
    - ◆ Delete a row or rows from the data: delete()

- ◆ Use a URI to identify the data you are reading or manipulating
  - ◆ URI: Uniform Resource Identifiers
  - ◆ URLs are a subset of URIs
  - ◆ `content://`  
   usually.some.base.package.name.followed.  
   by.some/terms
  - ◆ `content://` is the Content Provider Prefix
  - ◆ `usually.some.base.package.name.followed.  
by.some` : Content Authority
  - ◆ `/terms` : helps identify which exact data you are interested to access
- ◆ In case of reading from the ContentProvider, display the information in the UI
- ◆ Try making database operations off the main thread
- Creating your own Content Provider
  - Create a class that extends from the content provider
    - ◆ extends ContentProvider
  - implement the `onCreate()` function
  - Register the provider in the AndroidManifest file
    - ◆ use a `<provider>`
      - ◆ `<provider android:name=""  
  android:authorities=""  
  android:exported="" />`
      - ◆ `authorities` is usually the package name for keeping it unique
      - ◆ `name` is the complete package path to the extended content provider class
      - ◆ `exported` attribute can be true or false which determines whether it can be accessed by other applications or not
  - Define URIs that identify the Content Provider

and the different data that it can return

- Add these URIs to the Contract class
- Build a URIMatcher to match URI patterns to integers
- Implement the CRUD methods