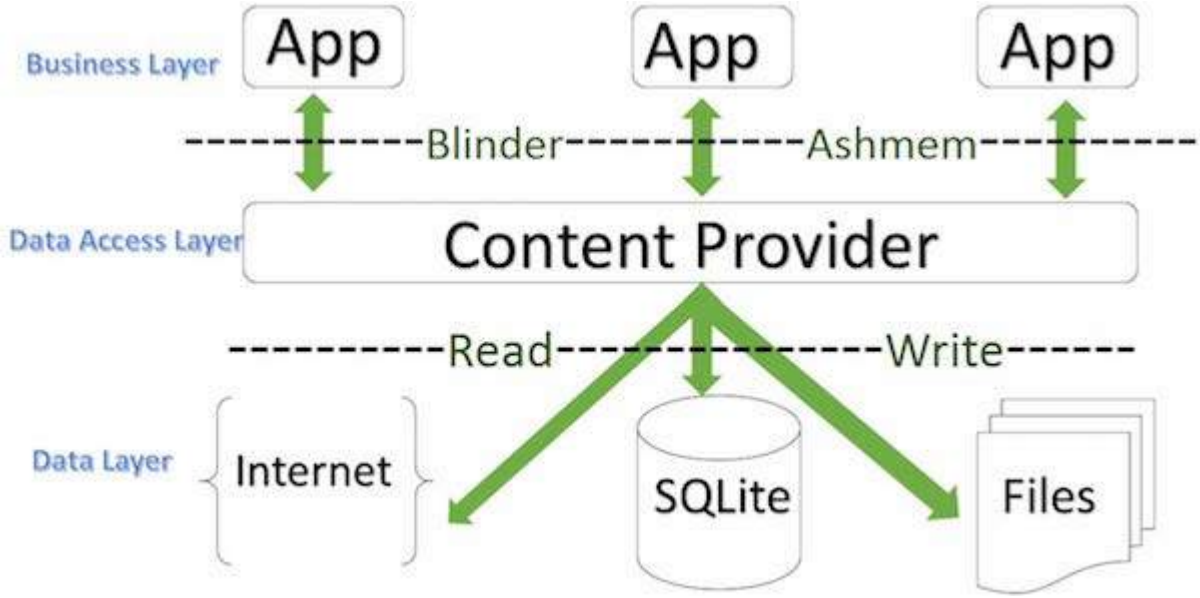
Content Providers

> A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the ContentResolver class. A content provider can use different ways to store its data and the data can be stored in a database, in files, or even over a network.



> sometimes it is required to share data across applications. This is where content providers become very useful.

Content providers let you centralize content in one place and have many different applications access it as needed. A content provider behaves very much like a database where you can query it, edit its content, as well as add or delete content using **insert()**, **update()**, **delete()**, and **query()** methods. In most cases this data is stored in an SQlite database.

A content provider is implemented as a subclass of ContentProvider class and must implement a standard set of APIs that enable other applications to perform transactions.

```
#Java
public class MyApplication extends ContentProvider {}
```

Content URIs

To query a content provider, you specify the query string in the form of a URI which has following format

```
<prefix>://<authority>/<data_type>/<id>
```

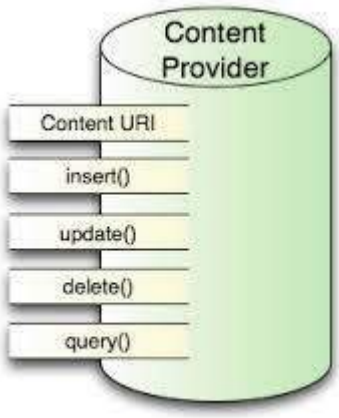
Part	Description
prefix	This is always set to content://
authority	This specifies the name of the content provider, for example contacts, browser etc. For third-party content providers, this could be the fully qualified name, such as dev.nxonxon.statusprovider
data_type	This indicates the type of data that this particular provider provides. For example, if you are getting all the contacts from the Contacts content provider, then the data path would be people and URI would look like this content://contacts/people
id	This specifies the specific record requested. For example, if you are looking for contact number 5 in the Contacts content provider then URI would look like this content://contacts/people/5.

Create Content Provider

This involves number of simple steps to create your own content provider.

- First of all you need to create a Content Provider class that extends the ContentProviderbaseclass.
- Second, you need to define your content provider URI address which will be used to access the content.
- Next you will need to create your own database to keep the content. Usually, Android uses SQLite database and framework needs to override **onCreate()** method which will use SQLite Open Helper method to create or open the provider's database. When your application is launched, the **onCreate()** handler of each of its Content Providers is called on the main application thread.
- Next you will have to implement Content Provider queries to perform different database specific operations.
- Finally register your Content Provider in your activity file using `<provider>` tag.

Here is the list of methods which you need to override in Content Provider class to have your Content Provider working



- **onCreate()** This method is called when the provider is started.
- **query()** This method receives a request from a client. The result is returned as a Cursor object.
- **insert()** This method inserts a new record into the content provider.
- **delete()** This method deletes an existing record from the content provider.
- **update()** This method updates an existing record from the content provider.
- **getType()** This method returns the MIME type of the data at the given URI.