Android Concepts

Content Providers

Introduction

Content providers store and retrieve data and make it accessible to all applications.

Android ships with a number of content providers for common data types (audio, video, images, personal contact information, and so on).

If you want to make your own data public, you have two options: You can create your own
content provider (a Content Provider subclass) or you can add the data to an existing provider —
if there's one that controls the same type of data and you have permission to write to it.

ContentResolver

When you want to access data in a content provider, you use the ContentResolver object in your application's context to communicate with the provider as a client.

You get a ContentResolver by calling getContentResolver() from within the implementation of an Activity or other application component

ContentResolver cr = getContentResolver();

Data Model

_ID	NUMBER	NUMBER_KEY	LABEL	NAME	TYPE
13	(425) 555 6677	425 555 6677	Kirkland office	Bully Pulpit	TYPE_WORK
44	(212) 555 1234	212 555 1234	NY apartment	Alan Vain	TYPE_HOME

Data Model Cont’d

_ID is a numeric column/field that uniquely identifies all records within the table

A query returns a Cursor object that can move from record to record and column to column to read the contents of each field.

URIs

Each content provider exposes a public URI (wrapped as a Uri object) that uniquely identifies its data set.

A content provider that controls multiple data sets (multiple tables) exposes a separate URI for each one.

NB: All URIs for providers begin with the string " content:// "

The content: scheme identifies the data as being controlled by a content provider.

URIs: Android defined

URI	Description
content://media/internal/images	URI return the list of all internal images on the device.
content://contacts/people/	URI return the list of all contact names on the device.

Every ContentResolver method takes the URI as its first argument.

Querying a Content Provider

You need at least three pieces of information to query a content provider:

- The URI that identifies the provider
- The name of the data fields you want to receive
- The data types of those fields

Making the query

To query a content provider you can either use ContentResolver.query() or Activity.managedQuery() method.

Both methods take the same set of arguments, and both return a Cursor object.

But managedQuery() has some few advantages over query():

- unloading itself when the activity pauses
- requerying itself when the activity restarts

Making the query Cont'd

Example;

```
Cursor cur  = getContentResolver().query(
        uriSMSURI, //content URI
        null, //columns to return for each row
        null, //selection criteria
        null, //selection criteria values
        BaseColumns._ID + " DESC LIMIT 40" //sort order
);
```

Making the query Cont'd

Parameters:

uri	The URI to query. This will be the full URI sent by the client; if the client is requesting a specific record, the URI will end in a record number that the implementation should parse and add to a WHERE or HAVING clause, specifying that _id value.
projection	The list of columns to put into the cursor. If null all columns are included.
selection	A selection criteria to apply when filtering rows. If null then all rows are included.
selectionArgs	You may include ?s in selection, which will be replaced by the values from selectionArgs, in order that they appear in the selection. The values will be bound as Strings.
sortOrder	How the rows in the cursor should be sorted. If null then the provider is free to define the sort order.

Resourceful Links

Contacts Provider

http://developer.android.com/guide/topics/providers/contacts-provider.html

Loading Contacts with Content Providers

https://github.com/codepath/android_guides/wiki/Loading-Contacts-with-Content-Providers