CSc 317
Content Providers

Benjamin Dicken

Announcements

- I’m back!
- Exam on Wednesday
- The Final Project
- A few more quizzes
- I need to update the links to the slides on the schedule
    - Can use D2L

Content Providers

- From the docs:
  - Help an application manage access to data stored by ***itself***, access data stored by ***other apps***, and provide a way to ***share data*** with other apps
  - Encapsulate the data, and provide mechanisms for defining data security
  - The standard interface that connects data in ***one process*** with code running in ***another process***

Content Providers

- You should have already gone over this with Tyler
  - Use a Content Provider
  - Display list of contacts
  - Click to see info about that contact

Content Resolver, General Steps

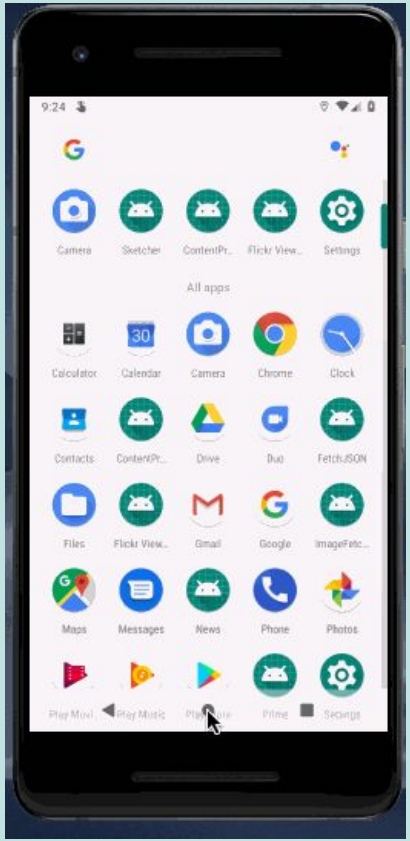
```
ContentResolver contentResolver = getActivity().getContentResolver();
Cursor query = contentResolver.query(
    uri,              // The URI describing source of data
    projection,       // What columns to grab
    selection,        // filter the rows, or null
    selectionArgs,    // replace placeholders in selection, or null
    sortOrder,        // Specify an ordering, or null
);
while (query.moveToNext()) {
    String data = query.getString(query.getColumnIndex("col_name"));
    // Display data, print data, etc
}
query.close()
```

The Goal

ICA

Get The Code

- Get starter code from the schedule
- Run the application
- Go settings, and give the application permission to view Contacts, Media, Messages



Generate Content for Testing

Using applications on the emulator

- Add several contacts with name, email, phone number (if you didn’t already)
- Open the messaging app and send several messages to the contacts
- Take a few photos using the default camera

Getting the Conversations / Threads

```
Cursor query = contentResolver.query(
        Uri.parse("content://mms-sms/conversations/"),
        new String[]{"_id", "thread_id", "address"},
        null, null, null);

while (query.moveToNext()) {
    String thread_id = query.getString(query.getColumnIndex("thread_id"));
    String address = query.getString(query.getColumnIndex("address"));
    String id = query.getString(query.getColumnIndex("_id"));
    // Display, print, etc
}
```

Get Messages for Specific Thread ID

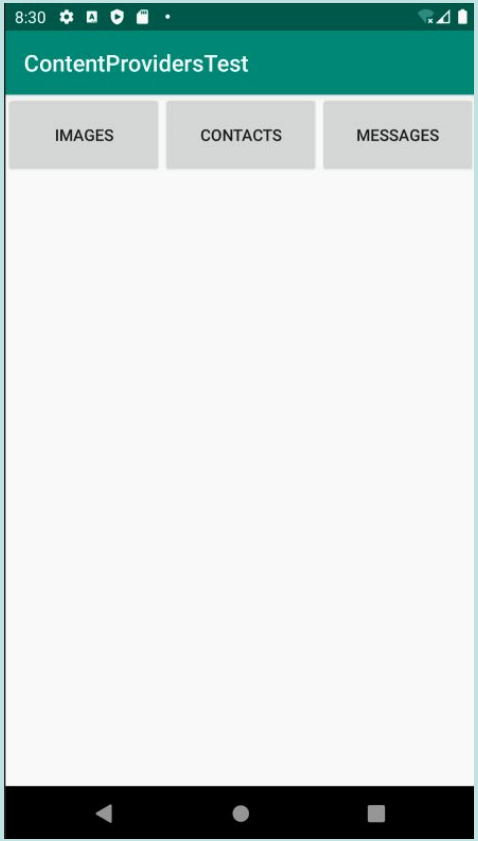
```
Cursor query = getContentResolver().query(
    Uri.parse("content://sms/"), null,
    "thread_id=" + thread_id,
    null, null);

while (query.moveToNext()) {
    String body = query.getString(query.getColumnIndexOrThrow("body"));
    // Display message, print, etc
}

query.close();
```

Display message threads

- Open up the starter code and address the TODO sections in **MessagesFragment**
- Get the list of message conversations to show
- When clicked on, should show the messages from that thread



Using a Content Provider (Media)

```
Cursor query = contentResolver.query(
        MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
        {MediaStore.Images.Media.DATA, MediaStore.Images.Media._ID},
        null, null, null);

for (int i = 0; i < query.getCount(); i++) {
    query.moveToPosition(i);
    int dataColumnIndex = query.getColumnIndex(MediaStore.Images.Media.DATA);
    String imageString = query.getString(dataColumnIndex);
    // Do something with imageString
}
```

Using a Content Provider (Media)

```
<!-- Also need to set this in application in XML -->
   <application
        . . . .
        android:requestLegacyExternalStorage="true">
```

Display message threads

- Open up the starter code and address the TODO sections in **ImagesFragment**
- Get the list of Images to show
- Don’t need to complete an action when an image is clicked