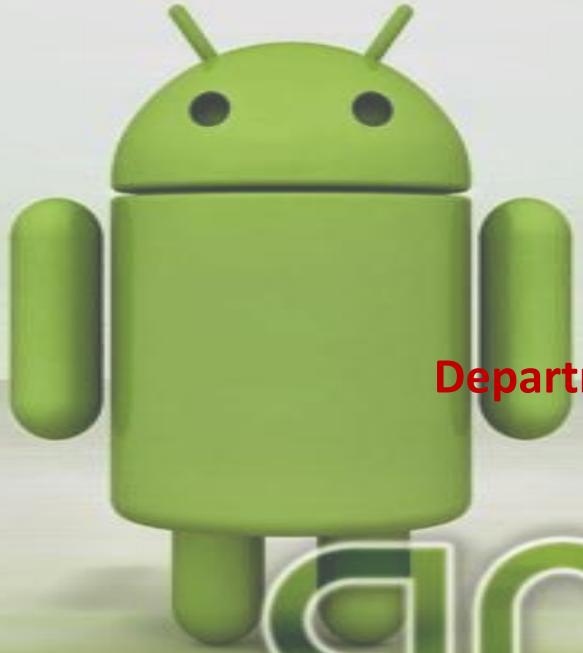
Android –Module 2



Mr.Jestin James M
Asst.Professor
Department of Computer Science
L.F College Guruvayoor

Structure of Android Content URIs

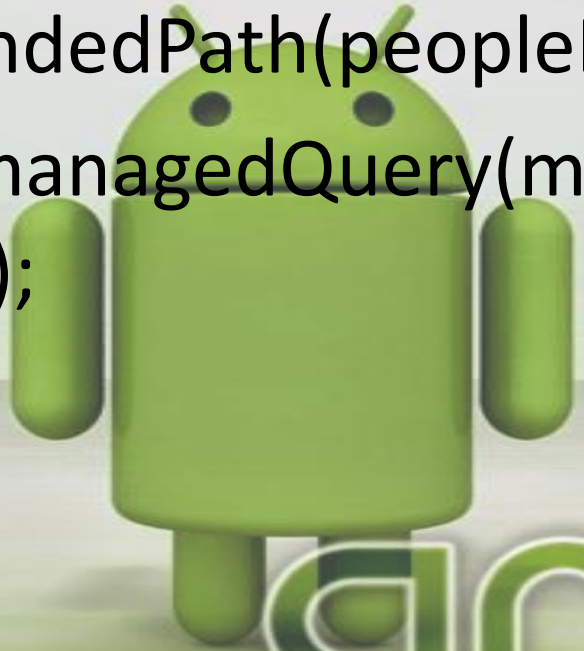
- retrieve data from a content provider invoke a URI
- It is a set of rows and columns represented by an Android cursor object.
- content://*/*/*
- content://com.google.provider.NotePad/notes/23

Reading Data Using URIs

- MediaStore.Images.Media.INTERNAL_CONTENT_URI
- MediaStore.Images.Media.EXTERNAL_CONTENT_URI
- ContactsContract.Contacts.CONTENT_URI
- content://media/internal/images
- content://media/external/images
- content://com.android.contacts/contacts/

Reading Data Using URIs

- Uri peopleBaseUri = ContactsContract.Contacts.CONTENT_URI;
- Uri myPersonUri = Uri.withAppendedPath(peopleBaseUri, "23");
- Cursor cur = managedQuery(myPersonUri, null, null, null);



Reading Data Using URIs

- **Listing 4–1.** ***Retrieving a Cursor from a Content Provider***
- import ContactsContract.Contacts;
- string[] projection = new string[] {
  Contacts._ID,
  Contacts.DISPLAY_NAME_PRIMARY
  };

Uri mContactsUri =
  ContactsContract.Contacts.CONTENT_URI;

Reading Data Using URIs

- Cursor managedCursor = managedQuery(mContactsUri,

  projection, //Which columns to return.

  null, // WHERE clause

  Contacts.DISPLAY_NAME_PRIMARY + " ASC"); // Order-by clause

Reading Data Using URIs

- http://developer.android.com/reference/android/provider/ContactsContract.Contacts.html
- every row returned has a default column
- called _id representing a unique ID for that row.

Using the Android Cursor

- A cursor is a collection of rows.
- use moveToFirst() before reading any data b'cos the cursor starts off positioned before the first row.
- You need to know the column names.
- You need to know the column types.
- All field-access methods are based on column number, so you must convert the column name to a column number first.

Using the Android Cursor

- The cursor is random
- Because the cursor is random, you can ask it for a row count.

Listing 4–2. *Navigating Through a Cursor Using a while Loop*

```
• if (cur.moveToFirst() == false)
{
   //no rows empty cursor
   return;
}
int nameColumnIndex =
   cur.getColumnIndex(Contacts.DISPLAY_NAME
   _PRIMARY)
```

Listing 4–2. *Navigating Through a Cursor Using a while Loop*

- String name = cur.getString(nameColumnIndex);

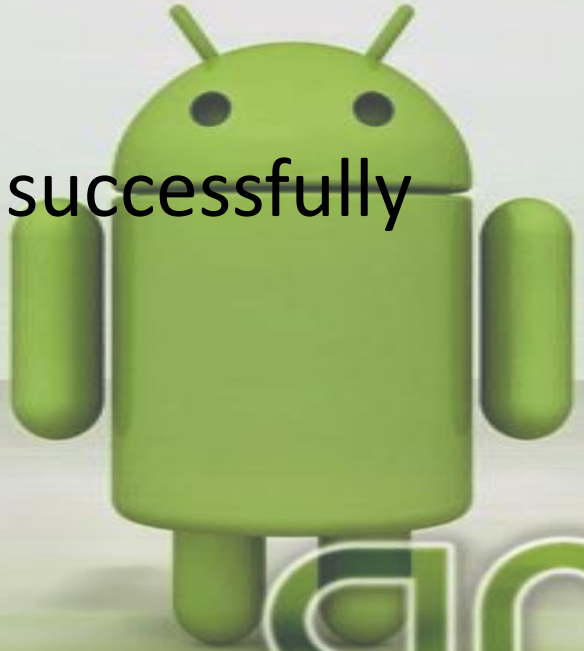
while(cur.moveToNext())

- {

//cursor moved successfully

//access fields

}



Using the Android Cursor

- isBeforeFirst()
- isAfterLast()
- isClosed()

Listing 4–3. *Navigating Through a Cursor Using a for Loop*

```
int nameColumn =
    cur.getColumnIndex(Contacts.DISPLAY_NAME
    _PRIMARY);
for(cur.moveToFirst();!cur.isAfterLast();cur.moveT
    oNext())
{
    String name = cur.getString(nameColumn);
}
```

Working with the where Clause

1. Through the URI
2. Through the combination of a string clause and a set of replaceable string-array arguments

Passing a where Clause Through a URI

- **Listing 4–4.** ***Passing SQL where Clauses Through the URI***
- String noteUri = "content://com.google.provider.NotePad/notes/23";
- Cursor managedCursor = someActivity.managedQuery(**noteUri,**
  projection, //Which columns to return.
  null, // WHERE clause
  null); // Order-by clause.