

# Android Components

## Android Smartphone Programming

University of Freiburg

**Matthias Keil**  
Institute for Computer Science  
Faculty of Engineering  
University of Freiburg

2. November 2015



**UNI  
FREIBURG**



- 1 Data Storage
- 2 Messages to the User
- 3 Background Work
- 4 App Widgets
- 5 Useful Widgets
- 6 Summary





- Many ways to store data of the application.
- Android provides support for SQLite databases<sup>[3]</sup>.
- Accessible for every class inside the application.
- No possibility to access from outside of application.
- Use *Cursor* object to loop through data inside the database.
- Easy to display cursor data inside an activity by creating an adapter and binding it to the activity, for example *ListAdapter* for *ListView*.



- Use method `execSQL` to execute one SQL statement.
- Exceptions: Do not use a `SELECT` statement or one that returns data.

```
1 try {
2     SQLiteDatabase db = openOrCreateDatabase("
        myDB", MODE_PRIVATE, null);
3     db.execSQL("CREATE TABLE IF NOT EXISTS
        myTable (Driver TEXT, Team TEXT);");
4     db.execSQL("INSERT OR IGNORE INTO myTable
        VALUES ('Glock', 'Toyota');");
5     ...
6 }
```





- Creation of Cursor object to access data from database

```
1 Cursor SQLiteDatabase.query(boolean
    distinct, String table, String[]
    columns, String selection, String[]
    selectionArgs, String groupBy,
    String having, String orderBy,
    String limit)
```

- Example

```
1 Cursor c = db.query(true, "myDB",
    values, null, null, null, null, null
    , null);
```



# Data Storage

Example: Accessing data from a table

University of Freiburg



```
1 try {
2     Cursor c = ...
3     c.moveToFirst();
4     String driverStr = c.getString(c.
        getColumnIndexOrThrow("Driver"));
5     String teamStr = c.getString(c.
        getColumnIndexOrThrow("Team"));
6     c.moveToNext();
7     ...
8 }
```





- Mostly used to share data between applications<sup>[2]</sup>.
- Many predefined Content Providers available, for example to access phone contacts.
- Provide mechanisms to define data security.
- Access of data through content URI.
  - Example: *people.CONTENT\_URI* to access the phone contacts.



# Data Storage

## Example: Usage of a Content Provider

University of Freiburg

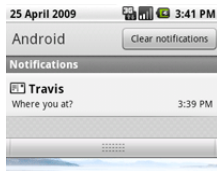


```
1 String[] proj = new String[] {People._ID,  
    People.NAME, People.NUMBER};  
2 try {  
3     Cursor c = managedQuery(People.CONTENT_URI,  
        proj, null, null, null);  
4     if (c == null) return;  
5     int name = c.getColumnIndexOrThrow(People.  
        NAME);  
6     if (!c.moveToFirst()) return;  
7     String nameString = c.getString(name);  
8     ...  
9 }
```





- Can be created using the *Notification* class.
- Adds an icon to the status bar and a message in the notifications window of the system<sup>[5]</sup>.
- Used to notify the user about an event from a background service, not from a visible application.
- Can be selected in the notifications window, which fires a predefined *intent* (message).



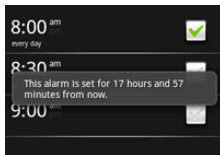
# Messages to the User

## Toast Notifications

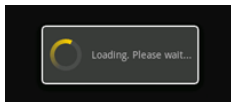
University of Freiburg



- *Toast*: Small message that pops up when shown<sup>[6]</sup>.
- User's current activity remains visible and interactive.
- Fades in and out.
- Does not provide any means for user interaction.
- Used to display short text messages.



- *Dialog*: Small window that appears in front of current Activity<sup>[4]</sup>.
- Used for notifications or interaction directly related with the application.
- Example usage scenarios:
  - Display of a progress bar till the application is updated.
  - Display a short message the user needs to confirm with *OK* or *Cancel*.

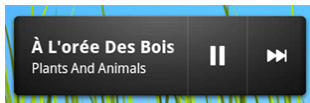




- Used to perform long-running operations in background<sup>[7]</sup>.
- Does not provide a user interface.
- Continues to run even when starting component is not active anymore.
- *Interprocess communication* (IPC) possible for interaction between component and service.
- Used for example for handling network transactions, play music, etc.



- Miniature application view that can be embedded in other application, for example home screen<sup>[1]</sup>.
- Receives periodic updates.
- Example usage: Displaying current song of music player as seen in picture below.





- *DatePicker* can be used to pick a date including year, month and day.
- *TimePicker* enables the user to specify a time in 24 hour or AM/PM mode.



- Persistent data storage can be archived with *SQLite databases*.
- Data can be shared between different applications using *Content Providers*.
- Messages to the user can be displayed as small text message through a *Toast*, a *Status Bar Notification* from a service or a *Dialog*, if user interaction is required.
- *Services* are used to perform background work.
- *App Widgets* can be placed on the home screen and receive periodic updates.





ANDROID DEVELOPERS.

App Widgets.

<http://developer.android.com/guide/topics/appwidgets/index.html>.



ANDROID DEVELOPERS.

Content Providers.

<http://developer.android.com/guide/topics/providers/content-providers.html>.



ANDROID DEVELOPERS.

Data Storage using Databases.

<http://developer.android.com/guide/topics/data/data-storage.html#db>.



ANDROID DEVELOPERS.

Notifications: Dialog.

<http://developer.android.com/guide/topics/ui/notifiers/index.html#Dialog>.



ANDROID DEVELOPERS.

Notifications: Status Bar.

<http://developer.android.com/guide/topics/ui/notifiers/index.html#StatusBar>.



ANDROID DEVELOPERS.

Notifications: Toast.

<http://developer.android.com/guide/topics/ui/notifiers/index.html#Toast>.



ANDROID DEVELOPERS.

Services.

<http://developer.android.com/guide/topics/fundamentals/services.html>.

