

Mobile Application Development (Design and)

16th class

Prof. Stephen Intille
s.intille@neu.edu

Today

- Q&A
- Plan for apps that need web service
- Checking for errors
- Look at example from SQLite
- Look at examples 2d graphics
- Tomorrow: 3D graphics
- MoBoogie: Creative Expression Through Whole Body Musical Interaction
 - Kevin Cheetham

Web service plan

- Key-value server
 - Put, get key-value pairs
 - Get list of keys
 - Clear all

- Files server
 - Put, get Images/binaries
 - Get list of all files
 - Clear all

More detail...

```
String key1 = "Akey1";
String key2 = "Akey2";
String value1 = "value1"; // What is max allowable length?
String value2= "name,email,password,height,weight"; // Needs to be ok for value to
have special characters such as commas. If any special characters are not allowed,
then Put command should check for that that case and throw an error

// Setup a new DB
Clear(TeamName, PASSWORD); // Clear all values from DB (only works for a given set of
preestablished TEAMNAMES)

boolean isSuccessful = Put(TeamName, PASSWORD, key1, value); //Check that value is
not greater than max allowable length. Return error if it is.
boolean isSuccessful = Put(TeamName, PASSWORD, key2, value);

String value = Get(TeamName, PASSWORD, key); // Returns null if invalid TEAMNAME or
PASSWORD or can't get data; returns "" if key not in DB, returns value if key found
```

Files...

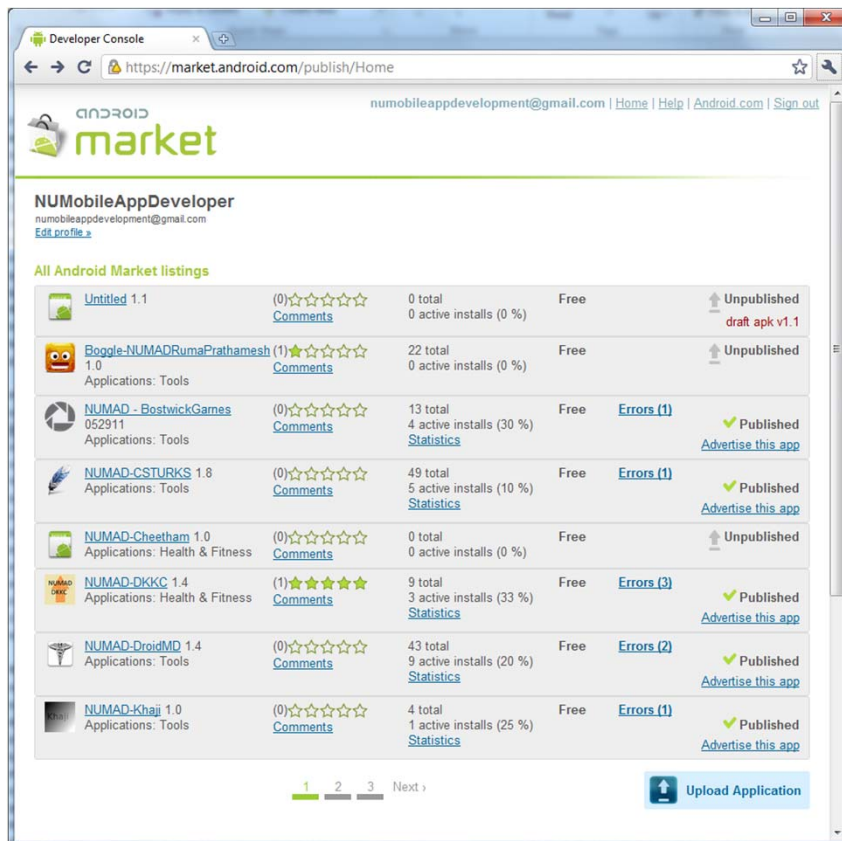
// More complicated:

```
boolean isSuccessful = put(TeAMNAME, PASSWORD, key, fileName);  
// fileName is name of a binary file (e.g. image) that is uploaded
```

```
boolean isSuccessful = get(TeAMNAME, PASSWORD, key,  
                           saveToFilename);  
// Returns path to the file after downloaded.
```

Checking for errors

- <https://market.android.com/publish/Home>



The screenshot shows the Android Market Developer Console interface. The user is logged in as 'NUMobileAppDeveloper'. The page displays a list of all Android Market listings for the user. The listings are as follows:

App Name	Version	Category	Rating	Total Installs	Active Installs	Percentage	Price	Status	Errors
Untitled	1.1		(0)☆☆☆☆☆	0 total	0 active	0%	Free	Unpublished	0
Boggle-NUMADRumaPrathamesh	1.0	Applications: Tools	(1)☆☆☆☆☆	22 total	0 active	0%	Free	Unpublished	0
NUMAD - BostwickGames	052911	Applications: Tools	(0)☆☆☆☆☆	13 total	4 active	30%	Free	Published	1
NUMAD-CSTURKS	1.8	Applications: Tools	(0)☆☆☆☆☆	49 total	5 active	10%	Free	Published	1
NUMAD-Cheetham	1.0	Applications: Health & Fitness	(0)☆☆☆☆☆	0 total	0 active	0%	Free	Unpublished	0
NUMAD-DKKC	1.4	Applications: Health & Fitness	(1)☆☆☆☆☆	9 total	3 active	33%	Free	Published	3
NUMAD-DroidMD	1.4	Applications: Tools	(0)☆☆☆☆☆	43 total	9 active	20%	Free	Published	2
NUMAD-Khaji	1.0	Applications: Tools	(0)☆☆☆☆☆	4 total	1 active	25%	Free	Published	1

At the bottom of the page, there is a progress indicator showing steps 1, 2, and 3, with 'Next' and an 'Upload Application' button.

SQLite (<http://www.sqlite.org>)

- Pros
 - Open source
 - Standards-compliant
 - Lightweight
 - Robust (hopefully)

- Cons
 - Loosely type columns

Syncing

- “Automatic” sync with web server database?
 - Typical scenario
- Recommended strategy
 - Keep data management/user simple on the phone
 - Keep complexity on the server
 - You don't want problems out of your control...

Basics

- ContentValues used to insert new rows into tables
- Queries are returned as Cursor objects
 - Pointers to result set within underlying data
 - Managed way of controlling position (row) in result set of a DB query
- `startManagingCursor` (`stopManagingCursor`)
 - Integrates Cursor lifetime into calling Activity's

Cursor class

- moveToFirst, moveToNext, moveToPrevious
- getCount
- getColumnIndexOrThrow (from name)
- getColumnName (from index)
- getColumnNames (in current cursor)
- moveToPosition (to row)
- getPosition (cursor position)

Helper classes

- Think through what you need and make the helper classes that will make use of DB go smoothly
 - Typing
 - Error checking
 - Syncing
 - Handle queries
 - Expose methods for creating, opening, closing
 - Publish DB constants

ContentProvider

- Generic, well-defined interface for using and sharing data
- Convention for URI:
 - Content://com.<CompanyName>.provider.<ApplicationName>/<DataPath>
 - Content://com.company.provider.myapp/elements (request for all values of type elements)
 - Content://com.company.provider.myapp/elements/5 (request for single, 5th element)

ContentProvider

- Generic, well-defined interface for using and sharing data
- Convention for URI:
 - Content://com.<CompanyName>.provider.<ApplicationName>/<DataPath>
 - Content://com.company.provider.myapp/elements (request for all values of type elements)
 - Content://com.company.provider.myapp/elements/5 (request for single, 5th element)

ContentProvider

- Typically exposing access to a SQLite DB
- But, can also expose access to any source of data (files, application instance variables)
- Use ContentResolver object to modify and query ContentProviders
- Query results returned as Cursors

ContentProvider

- Using query in ContentResolver, pass in:
 - URI of the ContentProvider you want to query
 - Projection that lists the columns you want in result set
 - A where clause that defines the rows to be returned (can use wildcards: ?)
 - An array of selection argument strings that replace wildcards (?)
 - A string that describes the order of the returned rows

Step through example

Graphics

- Two options
 - 2D graphics (android.graphics)
 - Meets most needs
 - Used in Sudoku/Boggle
 - 3D graphics using OpenGL
 - For games, rapid response

Color

- ARGB
 - 32-bit integer
 - Each value 256 values (8 bits)
- `int color = Color.BLUE`
- `int color = Color.argb(127, 255, 0, 0)`
- Or, can define in XML file

```
<color name="mycolor" >#7fff0000</color>
```

`int color = getResources().getColor(r.color.mycolor)`

Path

- Holds vector-drawing commands (lines, rectangles, curves)
- Easy way to do some neat stuff (e.g., text that hugs a path)

Drawable

- Bitmap
- NinePatch
- Shape (vector drawing based on Path)
- Layers
- States (focus states for buttons)
- Levels (shows one of a bunch of images)
- Scale (modifies size based on level)

Drawable

- Bitmap
- NinePatch
- Shape (vector drawing based on Path)
- Layers
- States (focus states for buttons)
- Levels (shows one of a bunch of images)
- Scale (modifies size based on level)

View Animations

- Easy way to get small animation into program
 - (Note: use animation sparingly...)
- Tween, Frame, Property (3.0+)

Tween Animations

- Tween
 - Series of simple transformations on view (position, size, rotation, transparency)
 - Simultaneous or sequential
 - <http://developer.android.com/guide/topics/graphics/view-animation.html>
- Example

Frame Animations

- Frame
 - Traditional frame-by-frame render

Property Animations

- 3.0 only, so avoid for now
- View animation only exposes a few aspects to animate and doesn't work on non-view objects
- Only animated view but properties of it don't change (button moving)