



Android



Creating an Earthquake Live Folder

2010.05.28

***Database Laboratory
Hanyang Univ.***

Live Folder



❖ Creating an Earthquake Live Folder

- In the following example you'll extend the Earthquake application again, this time to include a Live Folder that displays the magnitude and location of each quake.

Live Folder



- ❖ 1. Start by modifying the **EarthquakeProvider** class. Create a new static URI definition that will be used to return the Live Folder items.

```
public static final Uri LIVE_FOLDER_URI =  
    Uri.parse("content://com.paad.provider.earthquake/live_folder");
```

Live Folder



- ❖ 2. Modify the **uriMatcher** object and **getType** method to check for this new URI request.

```
private static final int LIVE_FOLDER = 3;

static {
    uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
    uriMatcher.addURI("com.paad.provider.Earthquake", "earthquakes", QUAKE_ID);
    uriMatcher.addURI("com.paad.provider.Earthquake", "earthquakes/#", QUAKE_ID);
    uriMatcher.addURI("com.paad.provider.Earthquake", "live_folder", LIVE_FOLDER);
}

@Override
public String getType(Uri uri) {
    switch (uriMatcher.match(uri)) {
        case QUAKE_ID:
            return "vnd.android.cursor.dir/vnd.paad.earthquake";
        case LIVE_FOLDER:
            return "vnd.android.cursor.item/vnd.paad.earthquake";
        default:
            throw new IllegalArgumentException("Unsupported URI: " + uri);
    }
}
```

Live Folder



- ❖ **3. Create a new hash map that defines a projection suitable for a Live Folder. It should return the magnitude and location details as the description and name columns respectively.**

```
static final HashMap<String, String> LIVE_FOLDER_PROJECTION;  
static {  
    LIVE_FOLDER_PROJECTION = new HashMap<String, String>();  
    LIVE_FOLDER_PROJECTION.put(LiveFolders._ID, KEY_ID + " AS " + LiveFolders._ID);  
    LIVE_FOLDER_PROJECTION.put(LiveFolders.NAME, KEY_DETAILS + " AS " + LiveFolders.NAME);  
    LIVE_FOLDER_PROJECTION.put(LiveFolders.DESCRPTION, KEY_DATE + " AS " + LiveFolders.DESCRPTION);  
}
```

Live Folder



- ❖ 4. Update the **query** method to apply the projection map from Step 4 to the returned earthquake query for Live Folder requests.

```
@Override
public Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sort) {
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();

    qb.setTables(EARTHQUAKE_TABLE);
    switch (uriMatcher.match(uri)) {
        case QUAKE_ID :
            qb.appendWhere(KEY_ID + "=" + uri.getPathSegments().get(1));
            break;
        case LIVE_FOLDER : qb.setProjectionMap(LIVE_FOLDER_PROJECTION);
            break;
        default : break;
    }
    [ ... existing query method ... ]
}
```

Live Folder



- ❖ 5. Create a new **EarthquakeLiveFolders** class that contains a static **EarthquakeLiveFolder** Activity.

```
package com.paad.earthquake;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.provider.LiveFolders;

public class EarthquakeLiveFolders extends Activity {
    public static class EarthquakeLiveFolder extends Activity {
    }
}
```

Live Folder



- ❖ 6. Add a new method that builds the Intent used to create the Live Folder. It should use the query URI you created in Step 1, set the display mode to list, and define the icon and title string to use. Also set the base Intent Intent to the individual item query from the Earthquake Provider:

```
private static Intent createLiveFolderIntent(Context context) {  
    Intent intent = new Intent();  
    intent.setData(EarthquakeProvider.LIVE_FOLDER_URI);  
    intent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_BASE_INTENT, new Intent(Intent.ACTION_VIEW, EarthquakeProvider.CONTENT_URI));  
    intent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_DISPLAY_MODE, LiveFolders.DISPLAY_MODE_LIST);  
    intent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_ICON, Intent.ShortcutIconResource.fromContext(context, R.drawable.icon));  
    intent.putExtra(LiveFolders.EXTRA_LIVE_FOLDER_NAME, "Earthquakes");  
    return intent;  
}
```


Live Folder



- ❖ 7. Override the **onCreate** method of the **EarthquakeLiveFolder** class to return the Intent defined in Step 6:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    String action = getIntent().getAction();
    if (LiveFolders.ACTION_CREATE_LIVE_FOLDER.equals(action))
        setResult(RESULT_OK, createLiveFolderIntent(this));
    else
        setResult(RESULT_CANCELED);
    finish();
}
```

Live Folder



- ❖ 8. Add the **EarthquakeLiveFolder** Activity to the application manifest, including an Intent Filter for the action **android.intent.action.CREATE_LIVE_FOLDER**:

```
<activity android:name=".EarthquakeLiveFolders$EarthquakeLiveFolder" android:label="All Earthquakes">
  <intent-filter>
    <action android:name="android.intent.action.CREATE_LIVE_FOLDER"/>
  </intent-filter>
</activity>
```

Live Folder



- ❖ You could expand this example by using the Earthquake Map Activity to display a specific quake when it's selected from the list.
- ❖ Start by adding an Intent Filter to the Earthquake Map Activity that listens for View actions on earthquake Content Provider data described by the Intent created in Step 6. Then improve the Activity to retrieve the location of the selected quake and center the map to that point.

