Branch for empty view

<https://github.com/udacity/Advanced_Android_Development/compare/2.00_Integration_Points_and_Error_Cases...2.01_Empty_Database>

after list view we add textbox for emptyview in xml layout

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.example.android.sunshine.app.ForecastFragment">

<ListView

style="@style/ForecastListStyle"

android:id="@+id/listview\_forecast"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:divider="@null" />

<!-- empty list -->

<TextView

android:id="@+id/listview\_forecast\_empty"

android:text="@string/empty\_forecast\_list"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center\_horizontal"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

/>

</FrameLayout>

Add this code in setemptyview in listview

mListView = (ListView) rootView.findViewById(R.id.listview\_forecast);

View emptyView = rootView.findViewById(R.id.listview\_forecast\_empty);

mListView.setEmptyView(emptyView);

mListView.setAdapter(mForecastAdapter);

string.xml

<string name="empty\_forecast\_list">No Weather Information Available</string>

https://github.com/udacity/Advanced\_Android\_Development/compare/2.01\_Empty\_Database...2.02\_Are\_We\_Offline

For more user friendly we check proper message for the empty list why list is empty no connectivity or poor connectivity etc..

First set permission in in manifest .

<!-- Permissions required to make our UI more friendly -->

<uses-permission

android:name="android.permission.ACCESS\_NETWORK\_STATE" />

add this function in our java class like utility class

/\*\*

\* Returns true if the network is available or about to become available.

\*

\* @param c Context used to get the ConnectivityManager

\* @return

\*/

static public boolean isNetworkAvailable(Context c) {

ConnectivityManager cm =

(ConnectivityManager)c.getSystemService(Context.CONNECTIVITY\_SERVICE);

NetworkInfo activeNetwork = cm.getActiveNetworkInfo();

return activeNetwork != null &&

activeNetwork.isConnectedOrConnecting();

}

This use in fragment …

/\*

Updates the empty list view with contextually relevant information that the user can

use to determine why they aren't seeing weather.

\*/

private void updateEmptyView() {

if ( mForecastAdapter.getCount() == 0 ) {

TextView tv = (TextView) getView().findViewById(R.id.listview\_forecast\_empty);

if ( null != tv ) {

// if cursor is empty, why? do we have an invalid location

int message = R.string.empty\_forecast\_list;

if (!Utility.isNetworkAvailable(getActivity()) ) {

message = R.string.empty\_forecast\_list\_no\_network;

}

tv.setText(message);

}

}

}

This function is set on load load finish function

@Override

public void onLoadFinished(Loader<Cursor> loader, Cursor data) {

mForecastAdapter.swapCursor(data);

if (mPosition != ListView.INVALID\_POSITION) {

// If we don't need to restart the loader, and there's a desired position to restore

// to, do so now.

mListView.smoothScrollToPosition(mPosition);

}

updateEmptyView();

}

String.xml

|  |
| --- |
|  |
| <string name="empty\_forecast\_list\_no\_network">No weather information available. The network is not available to fetch weather data.</string> |
|  |

Theses cases for sate of error define through annotation@

This constants show the different status

@Retention(RetentionPolicy.SOURCE)

@IntDef({LOCATION\_STATUS\_OK, LOCATION\_STATUS\_SERVER\_DOWN, LOCATION\_STATUS\_SERVER\_INVALID, LOCATION\_STATUS\_UNKNOWN})

public @interface LocationStatus {}

public static final int LOCATION\_STATUS\_OK = 0;

public static final int LOCATION\_STATUS\_SERVER\_DOWN = 1;

public static final int

LOCATION\_STATUS\_SERVER\_INVALID = 2;

public static final int

LOCATION\_STATUS\_UNKNOWN = 3;

This is use in syncadapter

/\*\*

\* Sets the location status into shared preference. This function should not be called from

\* the UI thread because it uses commit to write to the shared preferences.

\* @param c Context to get the PreferenceManager from.

\* @param locationStatus The IntDef value to set

\*/

static private void setLocationStatus(Context c, @LocationStatus int locationStatus){

SharedPreferences sp = PreferenceManager.getDefaultSharedPreferences(c);

SharedPreferences.Editor spe = sp.edit();

spe.putInt(c.getString(R.string.pref\_location\_status\_key), locationStatus);

spe.commit();

}

https://github.com/udacity/Advanced\_Android\_Development/compare/2.04\_Status\_Storage...2.05\_Show\_Users\_the\_Status