

User Interface >

Binding to Data with AdapterView

The <u>AdapterView</u> is a ViewGroup subclass whose child Views are determined by an <u>Adapter</u> that binds to data of some type. AdapterView is useful whenever you need to display stored data (as opposed to resource strings or drawables) in your layout.

<u>Gallery</u>, <u>ListView</u>, and <u>Spinner</u> are examples of AdapterView subclasses that you can use to bind to a specific type of data and display it in a certain way.

AdapterView objects have two main responsibilities:

- Filling the layout with data
- · Handling user selections

In this document

Filling the Layout with Data Handling User Selections

Related tutorials

Spinner

List View

Grid View

Filling the Layout with Data

Inserting data into the layout is typically done by binding the AdapterView class to an <u>Adapter</u>, which retrieves data from an external source (perhaps a list that the code supplies or query results from the device's database).

The following code sample does the following:

- 1. Creates a <u>Spinner</u> with an existing View and binds it to a new ArrayAdapter that reads an array of colors from the local resources.
- 2. Creates another Spinner object from a View and binds it to a new SimpleCursorAdapter that will read people's names from the device contacts (see Contacts.People).

```
// Get a Spinner and bind it to an ArrayAdapter that
// references a String array.
Spinner s1 = (Spinner) findViewById(R.id.spinner1);
ArrayAdapter adapter = ArrayAdapter.createFromResource(
    this, R.array.colors, android.R.layout.simple spinner item);
adapter.setDropDownViewResource(android.R.layout.simple spinner dropdown item);
s1.setAdapter(adapter);
// Load a Spinner and bind it to a data query.
private static String[] PROJECTION = new String[] {
       People. ID, People.NAME
Spinner s2 = (Spinner) findViewById(R.id.spinner2);
Cursor cur = managedQuery(People.CONTENT URI, PROJECTION, null, null);
SimpleCursorAdapter adapter2 = new SimpleCursorAdapter(this,
    android.R.layout.simple spinner item, // Use a template
                                          // that displays a
                                          // text view
    cur, // Give the cursor to the list adapter
    new String[] {People.NAME}, // Map the NAME column in the
                                         // people database to...
```

Note that it is necessary to have the People._ID column in projection used with CursorAdapter or else you will get an exception.

If, during the course of your application's life, you change the underlying data that is read by your Adapter, you should call notifyDataSetChanged(). This will notify the attached View that the data has been changed and it should refresh itself.

Handling User Selections

You handle the user's selection by setting the class's AdapterView.OnItemClickListener member to a listener and catching the selection changes.

```
// Create a message handling object as an anonymous class.
private OnItemClickListener mMessageClickedHandler = new OnItemClickListener() {
    public void onItemClick(AdapterView parent, View v, int position, long id)
    {
        // Display a messagebox.
        Toast.makeText(mContext,"You've got an event",Toast.LENGTH_SHORT).show();
    }
};

// Now hook into our object and set its onItemClickListener member
// to our class handler object.
mHistoryView = (ListView) findViewById(R.id.history);
mHistoryView.setOnItemClickListener(mMessageClickedHandler);
```

For more discussion on how to create different AdapterViews, read the following tutorials: <u>Hello Spinner</u>, <u>Hello ListView</u>, and <u>Hello GridView</u>.

← Back to User Interface ↑ Go to top

Except as noted, this content is licensed under <u>Apache 2.0</u>. For details and restrictions, see the <u>Content License</u>. Android 3.1 r1 - 17 Jun 2011 10:58

<u>Site Terms of Service</u> - <u>Privacy Policy</u> - <u>Brand Guidelines</u>