**ListView Example**

**Example App**: “ListViewExample.zip”

**Purpose of this Example:**

* Learn how to get items from a list to display a ListView
* Later you will need to get items from a Backendless database to display in a ListView. It’s more complicated to do that with external data, so for this example we keep it simple by creating some Comment objects and making our own ArrayList<Comment>

**ListView Components**

1. Create an ArrayList<> with some objects in it. This will be the data that is eventually displayed in your ListView. In the example app, two Comment objects are created and added into an ArrayList<Comment>; you can use this as a guide.
2. Add a ListView to your main activity’s XML file. Be sure to give it an “id” attribute.
3. Declare your ListView at the top of your class, then instantiate it in your onCreate() method.
4. Create an XML layout file that defines how each item will be displayed within the list. It can be as simple or as complex as you like. The simplest would contain only a single TextView, but it could also include things like Edit and Delete buttons, extra TextViews to display other information from the Comment object (id number, author email, etc), and so on.
5. Create a custom ArrayAdapter<Comment> class. The example app calls it “CommentListAdapter<Comment>.” It must extend the ArrayAdapter class.

The ArrayAdapter’s job is to build the individual views for each item in the ArrayList. It extracts data from each comment object, then uses it to inflate a view using the XML layout you created in the previous step.

Make sure that your ArrayAdapter references the right layout file in its inflate() method. This is the mini-layout you created in the previous step.

The “position” variable represents the item number in your ArrayList. For example, if position = 3, that’s like myArrayList.get(3), or item number 3 in the ArrayList. Notice the line that says “Comment comment = getItem(position);” 🡪 this command stores one of the Comment objects in your ArrayList in an object called “comment,” which allows you to access its data using comment.getMessage() and so on.

The overridden getView() method is similar to the onCreate() method for an Activity or Fragment. This is where you instantiate your View objects (such as TextViews) and set them up, for example using setText().

When the ArrayAdapter is used, its getView() method will be called once for every object in the ArrayList.

1. Create an instance of your custom array adapter class in the onCreate() method. Look at the example and notice that its first argument is the activity (“this”) and the second one is the ArrayList containing the data.
2. Set your ListView object to display the data from your ArrayAdapter using the ListView’s setAdapter() method. If everything has gone well, you can now run your app and see your data displayed in the ListView.