How to search list view

Introduction

You have a long list, you do not want to skim through the entire list to find the item you want, solution? Include an EditText search box at the top of the screen, we will filter the list using the input text and display result.

Requirements

```
import android.text.Editable;
import android.text.TextWatcher;
```

There are two types of configuration for this search approach:

- Override toString() of the list item object
- Program your own search method in the onTextChanged()

Advantage and disadvantage

Advantages:

- Don't need any database
- Default APIs

Disadvantages:

- Can only query using text (string data type) by default

Steps and code snapshot

Basically, the text watcher will use a filter object of the current activity to redisplay the screen. This filter object will search using the toString() of the list item object. Thus, overriding the toString() to return the desired String for query is the recommended way.

For example, we have a list of companies with the following attributes: name, tax number, address. You want to search the list using company name, then override Company.toString() to return Company.name.

If you want to customize the search filter to do fancier things (not recommended), you will have to either find a way to make your own version of Filter object for the current activity or handle everything from GUI to functions in the onTextChanged() without the help of built-in API (again, not recommended).

```
public void onCreate(Bundle b) {
    super.onCreate(b);

    // initialize search box and list, our search box is at the top
    EditText inputSearch = (EditText)
findViewById(R.id.eventList_inputSearch);
    ListView list = (ListView) findViewById(R.id.eventList_list);

    // replace generic type (Object in this example) with your object of
choice
    // this array list holds the list item data that will be displayed
    ArrayList<Object> array = new ArrayList<Object>();
    ArrayAdapter<Object> adapter = new ArrayAdapter<Object>(this, array);
    list.setAdapter(adapter);

    inputSearch.addTextChangedListener(new TextWatcher() {

        // need to override this function
        @Override
        public void onTextChanged(CharSequence cs, int arg1, int arg2,
                int arg3) {
            // if you override toString() of the list item object,
            // include this line
            //
            // CurrentActivity is the class name of the activity holding the
            // list view
            CurrentActivity.this.adapter.getFilter().filter(cs);

            // else write your own search engine and visual effect
            // handling here
        }

        @Override
        public void beforeTextChanged(CharSequence arg0, int arg1,
                int arg2, int arg3) {
            // handle fancy visual effect or other function if you want
        }

        @Override
        public void afterTextChanged(Editable arg0) {
            // handle fancy visual effect or other function if you want
        }

    });
}
```