

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

☰ 目录视图

☰ 摘要视图

PDF 浏览

从创业到再就业，浅谈对程序员职业生涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Android SearchView结合Filter实现列表检索功能使用总结

标签： android SearchView filter 列表检索

2016-01-10 19:12 328人阅读 评论(0) 收藏 举报

☰ 分类： 移动开发 (38) ▾

版权声明：本文为博主原创文章，未经博主允许不得转载。



下面贴出主要代码：

```
[java]
01. <span style="font-size:14px;">
02. public class MainActivity extends FragmentActivity implements OnClickListener{
03.     public static SearchView mSearchView = null;
04.     public static ListView lv = null;
05.     private LazyAdapter adapter = null;
06.
07.     @Override
08.     protected void onCreate(Bundle savedInstanceState) {
09.         super.onCreate(savedInstanceState);
10.         setContentView(R.layout.activity_main);
11.         lv = (ListView) this.findViewById(R.id.lv);
12.         adapter = new LazyAdapter(this, list);
13.         lv.setAdapter(adapter);</span><span style="font-size:14px;">
```

```

14.
15.         mSearchView = (SearchView) findViewById(R.id.srv);
16.         mSearchView.setIconifiedByDefault(false);
17.         mSearchView.setSubmitButtonEnabled(false);
18.     mSearchView.setFocusable(true);
19.     mSearchView.setFocusableInTouchMode(true);
20.     /*
21.     mSearchView.requestFocus();
22.     mSearchView.clearFocus();
23.     mSearchView.onActionViewExpanded();
24.     */
25.     mSearchView.setOnQueryTextListener(new OnQueryTextListener() {
26.         @Override
27.         public boolean onQueryTextSubmit(String query) {
28.             return false;
29.         }
30.         @Override
31.         public boolean onQueryTextChange(String newText) {
32.             ListAdapter adapter = lv.getAdapter();
33.             if(adapter instanceof Filterable){
34.                 Filter filter = ((Filterable)adapter).getFilter();
35.                 if(newText==null || newText.length()==0){
36.                     filter.filter(null);
37.                 }else{
38.                     filter.filter(newText);
39.                 }
40.             }
41.             return true;
42.         }
43.     });
44. }
45.
46. @Override
47. protected void onResume() {
48.     mSearchView.setFocusable(true);
49.     mSearchView.setFocusableInTouchMode(true);
50.     // mSearchView.requestFocus(); //获取焦点
51.     super.onResume();
52. }
53.
54. @Override
55. public boolean onCreateOptionsMenu(Menu menu) {
56.     getMenuInflater().inflate(R.menu.main, menu);
57.     return true;
58. }
59. @Override
60. public boolean onOptionsItemSelected(MenuItem item) {
61.     return super.onOptionsItemSelected(item);
62. }
63. }
64.
65. </span>

```

[java]

```

01. <span style="font-size:14px;">
02. public class LazyAdapter extends BaseAdapter implements Filterable{
03.     private Activity activity;
04.     private static LayoutInflater inflater=null;
05.     public ImageLoader imageLoader;
06.     private List<Map<String, String>> list = null;
07.     private List<Map<String, String>> values = null;
08.     private MyFilter mFilter;
09.     private Object mLock = new Object();
10.
11.     public LazyAdapter(Activity a, List<Map<String, String>> list) {
12.         this.activity = a;
13.         this.list = list;
14.         inflater = (LayoutInflater)activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
15.         imageLoader=new ImageLoader(activity.getApplicationContext());
16.     }
17.     public int getCount() {
18.         return list.size();
19.     }
20.     public Object getItem(int position) {
21.         return position;
22.     }
23.     public long getItemId(int position) {
24.         return position;

```

```

25.     }
26.     @SuppressWarnings("InflateParams")
27.     public View getView(int position, View convertView, ViewGroup parent) {
28.         View vi=convertView;
29.         if(convertView==null){
30.             vi = inflater.inflate(R.layout.item_layout, null);
31.         }
32.         TextView text = BaseViewHolder.get(vi, R.id.title);
33.         TextView fu_text = BaseViewHolder.get(vi, R.id.fu_title);
34.         ImageView image=BaseViewHolder.get(vi, R.id.iv);
35.         text.setText(list.get(position).get("title"));
36.         fu_text.setText(list.get(position).get("fu_title"));
37.         imageLoader.DisplayImage(list.get(position).get("src"), image);
38.         return vi;
39.     }
40.
41.     @Override
42.     public Filter getFilter() {
43.         if(mFilter==null){
44.             mFilter = new MyFilter();
45.         }
46.         return mFilter;
47.     }
48.     class MyFilter extends Filter{
49.         @SuppressWarnings("DefaultLocale")
50.         @Override
51.         protected FilterResults performFiltering(CharSequence prefix) {
52.             FilterResults results = new FilterResults();
53.             if(values==null){
54.                 synchronized (mLock) {
55.                     values = new ArrayList<Map<String, String>>(list);
56.                 }
57.             }
58.             if(prefix==null || prefix.length()==0){
59.                 synchronized (mLock) {
60.                     List<Map<String, String>> list1 = new ArrayList<Map<String, String>>(values);
61.                     results.values = list1;
62.                     results.count = list1.size();
63.                 }
64.             }else{
65.                 String prefixString = prefix.toString().toLowerCase();
66.                 List<Map<String, String>> values1 = values;
67.                 int count = values1.size();
68.                 List<Map<String, String>> newValues = new ArrayList<Map<String, String>>(count);
69.                 for(Map<String, String> value : values1){
70.                     String title = value.get("title").toLowerCase();
71.                     if(title.indexOf(prefixString) != -1){
72.                         newValues.add(value);
73.                     }
74.                 }
75.                 results.values = newValues;
76.                 results.count = newValues.size();
77.             }
78.             return results;
79.         }
80.         @SuppressWarnings("unchecked")
81.         @Override
82.         protected void publishResults(CharSequence constraint,
83.             FilterResults results) {
84.             list = (List<Map<String,String>>) results.values;
85.             XinwenFragment.list = list;
86.             if(results.count>0){
87.                 notifyDataSetChanged();
88.             }else{
89.                 notifyDataSetInvalidated();
90.             }
91.         }
92.     }
93. }
94. </span>

```