

# 视频在线搜索设计与实现

---

**By Long Luo**

**@2014.06.27**

## Agenda

---

1. 在线搜索现状及发展
2. UI设计及实现
3. 代码逻辑设计及实现
4. 问题及解决

## 一. 在线搜索

---

之前搜索页面的一些缺陷：

1. 具体实现位于VideoListActivity中，一方面会造成VideoListActivity代码过于庞大臃肿，另外一方面不便于后续功能扩展，结构不清晰；
2. 依赖了大量系统控件，不便于后续解耦及界面定制；
3. 今后搜索界面会参考第三方视频应用的实现，之前不便于增加搜索记录，或搜索独立出来，用于搜索本地视频，甚至将此搜索移植应用于其他应用中；

### 1. 在线搜索实现效果

在线搜索因为是和第三方合作，涉及到很多网络相关的操作，简单来说就是利用HTTP协议向相关接口发起一次网络请求，服务器如果返回了正确的响应，App会解析服务器返回的内容，并展示出来。

#### a. 热词界面

热词界面是当搜索文本框文字为空时会弹出热词界面，会展示最近一段时间内搜索频率很高的词语。一方面可以节省大家输入文字，另外一方面你也可以了解当前的一些热点。

当你点击列表中的某个热词时，就会发起一次以此为关键词的搜索。



**b. 关联词**

当搜索文本输入框含有文字时，会获取当前输入文字，以此为关键词获取网络的一些联想词，可以点击此联想词发起一次搜索。



c. 搜索结果分类浏览

发起一次搜索之后，如果得到了服务器的正确响应，而且确实有相关视频内容。那么我们会将搜索结果展示在手机页面上。

搜索到的结果可以分不同频道浏览，会根据具体内容进行动态变化，有的可能有10几个频道，有的也就1个频道。频道页面可以滑动浏览，也可以选择在顶部页面选中或者滑动。

分类浏览时，第一个展示的页面是搜索到的全部视频内容，之后的会根据结果动态变化。

如下图所示：

15:43

Q 晓松奇谈


全部

娱乐

片花

综艺


纪录片



晓松奇谈之世界杯之拉美自相...

播放源:爱奇艺


主演:



晓松奇谈之世界杯之西班牙折...

播放源:爱奇艺


主演:



晓松奇谈之“旗”妙物语世界杯...

播放源:爱奇艺


主演:



晓松奇谈之“旗”妙物语世界杯...

播放源:爱奇艺

主演:



《晓松奇谈》会说话的广告牌 ...

播放源:爱奇艺

分频道浏览:



d. 语音搜索

语音搜索图标只有当搜索框里文字为空才会出现，否则出现搜索图标。

点击语音搜索图标将会启动VoiceSearch这个apk，然后你可以说话，如果被正确识别之后，会发起一次搜索，并将结果展示出来。



e. 语音搜索结果



## 二. 在线搜索UI设计及实现

任何功能都离不开UI和代码。在此我们先讨论在线搜索界面的UI设计及具体实现：

### 1. SearchBar

SearchBar即为顶部的搜索栏，包括返回、编辑框、搜索按钮、语音按钮等。假如采用标准SDK，还需要加上一个清除全部文字按钮。

```
<RelativeLayout
    android:id="@+id/searchBar"
    android:layout_width="match_parent"
    android:layout_height="@dimen/searchBarHeight"
    android:layout_alignParentTop="true"
    android:background="@drawable/searchbar_bg"
    android:focusable="true"
    android:focusableInTouchMode="true"
    android:gravity="center" >

    <RelativeLayout
        android:id="@+id/searchBack"
        android:layout_width="wrap_content"
        android:layout_height="match_parent" >
```

```
<ImageView
    android:id="@+id/searchBackButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:background="@drawable/phone_search_back_arrow"
    android:clickable="true"
    android:contentDescription="@null"
    android:focusable="true" />
</RelativeLayout>

<RelativeLayout
    android:id="@+id/searchSubmitLayout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:layout_marginRight="@dimen/searchBar_SearchMarginRight" >

    <ImageView
        android:id="@+id/searchVoiceSubmit"
        android:layout_width="@dimen/searchBar_VoiceSearchButtonWidth"
        android:layout_height="@dimen/searchBar_VoiceSearchButtonHeight"
        android:layout_centerVertical="true"
        android:background="@drawable/video_search_voice_bg"
        android:visibility="gone" />

    <ImageView
        android:id="@+id/searchSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:background="@drawable/video_search_submit_bg" />
</RelativeLayout>

<RelativeLayout
    android:id="@+id/searchInputLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_centerInParent="true"
    android:layout_marginLeft="@dimen/searchBar_marginLeft"
    android:layout_toLeftOf="@id/searchSubmitLayout"
    android:layout_toRightOf="@id/searchBack" >

    <com.oppo.widget.OppoEditText
        android:id="@+id/searchKeyword"
        android:layout_width="match_parent"
        android:layout_height="@dimen/searchBar_EditTextHeight"
        android:layout_centerInParent="true"
        android:layout_centerVertical="true"
        android:background="@drawable/video_search_input_bg"
        android:ellipsize="end"
        android:hint="@string/search_hit"
        android:inputType="text"
        android:paddingLeft="@dimen/searchBar_EditTextPaddingLeft"
        android:paddingRight="@dimen/searchBar_EditTextPaddingRight"
        android:singleLine="true"
        android:textSize="14sp"
        oppo:quickDelete="true" />

</RelativeLayout>
</RelativeLayout>
```

## 2. 热词和关联词

实现热词和关联词需要ListView及相关缓冲、空瓶动画，使用OPPO SDK控件实现。

## 3. 分类筛选页面

分类筛选页面是一个难点，为了实现这个效果，使用了2种方案，但第一种方案页面无法滑动，最后选择可滑动页面方案。

### <1>. 水平ListView和ListView实现方案



这种方案是参考了iQiyi的实现方案，如下图所示：



iQiyi实现

分类可滑动使用了一个HorizontalListView实现，搜索结果使用ListView实现，优点：

- 1. 每个频道显示可以自定义，可以不仅仅显示频道名称，后续扩展方便，比如添加具体视频数字等；
- 2. 代码逻辑简单，仅需要添加ListView点击实现接口，启动搜索，展示结果。

缺点： 不同频道页面不可以滑动切换，无法满足UE需求。

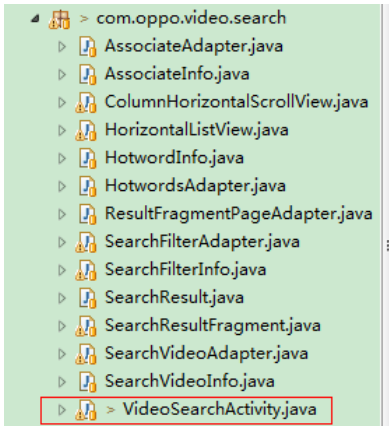
<2>. 滑动实现方案

按照iQiyi方案实现之后，由于需求变更。必须实现分类页面滑动切换效果，于是就有了第二种方案。

- 1. 分类频道使用HorizontalScrollView实现， ScrollView中添加相应的View，展示搜索分类；
- 2. 由于页面可滑动，需要ViewPager和Fragments List，在滑动时，启动相关的fragment，显示相关内容；
- 3. 每个fragment需要ListView及相关的一些控件，发起搜索及相关实现；

三. 在线搜索代码逻辑

1. 搜索类



iQiyi搜索接口：

<http://iface.iqiyi.com/api/searchIface?>

key=xxx&id=7f15c6eafc&type=xml&version=1.0&searchtype=2&pagesize=21&page\_number=1&keyword=%E4%B8%96%E7%95%8C%E6%9D%AF

在某个频道内搜索：

<http://iface.iqiyi.com/api/searchIface?>

key=xxx&id=7f15c6eafc&type=xml&version=1.0&searchtype=2&pagesize=21&pagenumber=1&keyword=%E4%B8%96%E7%95%8C%E6%9D%AF&categoryid

## 2. 搜索结果频道分类

```
<weights>
  <weight>
    <category_name>
      <![CDATA[ 全部 ]]>
    </category_name>
    <category_id>0</category_id>
    <count>86</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 娱乐 ]]>
    </category_name>
    <category_id>7</category_id>
    <count>2</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 音乐 ]]>
    </category_name>
    <category_id>5</category_id>
    <count>1</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 综艺 ]]>
    </category_name>
    <category_id>6</category_id>
    <count>11</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 时尚 ]]>
    </category_name>
    <category_id>13</category_id>
    <count>2</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 体育 ]]>
    </category_name>
    <category_id>17</category_id>
    <count>66</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 搞笑 ]]>
    </category_name>
    <category_id>22</category_id>
    <count>3</count>
  </weight>
  <weight>
    <category_name>
      <![CDATA[ 财经 ]]>
    </category_name>
    <category_id>24</category_id>
    <count>1</count>
  </weight>
</weights>
```

我们解析这个字段中的内容，就可以获取相关分类频道及其结果数量。

设计搜索的数据结构如下：

```
public class SearchResult {
    public ArrayList<SearchFilterInfo> weightList;
    public ArrayList<SearchVideoInfo> searchVideoList;
}
```

### 3. 搜索结果页面

每一个分类页面都是一个fragment，全部页面是一个fragments List来管理，启动时我们需要初始化，将一些必要的数据传递给各个fragment。

```
private void initFragment() {
    int count = mSearchResult.weightList.size();

    MyLog.d(TAG, "initFragment, count=" + count);

    for (int i = 0; i < count; i++) {
        Bundle data = new Bundle();
        int channelId = StringUtils.toInt(mSearchResult.weightList.get(i).category_id, 0);
        data.putString("channelTitle", mSearchResult.weightList.get(i).category_name);
        data.putString("keyword", mKeyword);
        data.putInt("channelId", channelId);
        SearchResultFragment newFragment = new SearchResultFragment();
        newFragment.setArguments(data);
        MyLog.d(TAG, "bundle=" + data + ",id=" + channelId + ",keyword=" + mKeyword);
        fragments.add(newFragment);
    }

    ResultFragmentPagerAdapter mAdapter = new ResultFragmentPagerAdapter(
        getSupportFragmentManager(), fragments);

    mSearchViewPager.setVisibility(View.VISIBLE);
    mSearchViewPager.setAdapter(mAdapter);
    mSearchViewPager.setOnPageChangeListener(pageListener);
    mAdapter.notifyDataSetChanged();
}
```

在不同搜索结果页面进行切换时，实现一个监听器，获取当前选中页面：

```
public OnPageChangeListener pageListener = new OnPageChangeListener() {

    @Override
    public void onPageScrollStateChanged(int arg0) {
        // MyLog.d(TAG, "onPageScrollStateChanged, arg0=" + arg0);
    }

    @Override
    public void onPageScrolled(int arg0, float arg1, int arg2) {
        // MyLog.d(TAG, "onPageScrolled, ");
    }

    @Override
    public void onPageSelected(int position) {
        MyLog.d(TAG, "onPageSelected, pos=" + position);
        mSearchViewPager.setCurrentItem(position);
        selectTab(position);
    }
};
```

在具体页面实现中，我们需要获取相关的参数及绘制页面：

```
private void getArgs() {

    Bundle args = getArguments();
    mChannelTitle = args != null ? args.getString("channelTitle") : "";
    mChannelId = args.getInt("channelId");
    mKeyword = args != null ? args.getString("keyword") : "";

    mLocalString = "title="+mChannelTitle+",id="+mChannelId+",mKeyword="+mKeyword;

    MyLog.d(TAG, "onCreate, " + mLocalString);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

    MyLog.d(TAG, "onCreateView, " + mLocalString);

    View view = inflater.inflate(R.layout.search_result_fragment, null);

    mEmptyResultView = (OppoEmptyBottle) view.findViewById(R.id.empty_result_view);
    mEmptyResultView.setMessage(R.string.search_no_result);

    mLoadingProcessView = mLayoutInflater.inflate(R.layout.loading_process_view, null);
    loadingProgress = (OppoProgressBar) view.findViewById(R.id.loading_process);

    mResultList = (OppoListView) view.findViewById(R.id.search_result_list);

    mResultList.setOnItemClickListener(mItemClickListener);
    mResultList.setOnScrollListener(mOnScrollListener);

    init();
    initData();

    return view;
}
```

## 四. 问题及解决

在完成这个需求过程中，也遇到了一些问题，不过最终还是都得以解决了。在这里，挑选几个典型问题来说明下，仅供大家后续参考：

### 1. 无法输入中文问题

搜索栏文本输入框有2种输入方式：

1. 直接在编辑框中输入文字；
2. 选中列表中的热词或者关联词，填充编辑框，启动搜索；

最开始，在相应的mTextWatcher和列表点击中使用了setKeywords()方法来实现编辑框的文字输入。

```

mTextWatcher = new TextWatcher() {

    @Override
    public void afterTextChanged(Editable s) {
        MyLog.d(TAG, "afterTextChanged:s=" + s);

        mSearchKeyword.setMaxWidth(mSearchKeyword.getWidth());

        mKeyword = s.toString();
        setKeyWords(mKeyword);

        if (StringUtils.isEmptyStr(s.toString())) {
            startQueryHotWords();
        } else {
            startQueryAssWords(VideoUtils.encodeUTF8(mKeyword));
        }
    }

    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {

    }
};

```

但是在手动输入文字中，**setKeywords()**方法由于需要兼顾列表输入文字方法，需要先将**mTextWatcher**remove，然后**setText()**，再添加**mTextWatcher**，这样就造成了每输入一个字符都会在编辑框中显示，就无法输入中文了。

```

mSearchKeyword.removeTextChangedListener(mTextWatcher);
mSearchKeyword.setText(word);
mSearchKeyword.setSelection(word.length());
mSearchKeyword.addTextChangedListener(mTextWatcher);

```

解决方法：

再建一个**setListSearchWords(String word)**方法，用于列表选词，问题得以解决。

```

public void setListSearchWords(String word) {
    MyLog.d(TAG, "setListSearchWords, word=" + word);

    mKeyword = word;

    if ((mSearchKeyword != null) && (word != null)) {
        mSearchKeyword.removeTextChangedListener(mTextWatcher);
        mSearchKeyword.setText(word);
        mSearchKeyword.setSelection(word.length());
        mSearchKeyword.addTextChangedListener(mTextWatcher);

        if (StringUtils.isEmptyStr(word)) {
            mSearchVoiceSubmit.setVisibility(View.VISIBLE);
            mSearchSubmit.setVisibility(View.GONE);
        } else {
            mSearchVoiceSubmit.setVisibility(View.GONE);
            mSearchSubmit.setVisibility(View.VISIBLE);
        }
    }
}

```

## 2. 视频异常退出问题

在实际中遇到了一些异常退出问题：

**<1>. 滑动页面异常退出**

原因是没有从服务器获取到正确的搜索视频结果，使用Message传递时，虽然message.obj不为空，但搜索结果为空，导致退出。

解决方案：

增加相应的空指针判断。

**<2>. 选择列表一个热词同时按下返回键，视频退出**

原因是Activity在onDestory()中，销毁了对应的fragments List，但是在此之前已经启动搜索，搜索结果通过Handler，绘制页面，但fragments已经为空，导致出现空指针。

解决方案：

增加相应的空指针判断。

***Created By Long Luo at 2014/6/27 20:09:59***

***Completed By Long Luo at 2014/7/2 16:39:54***