



# آموزش برنامه نویسی اندروید در محیط اندروید استودیو

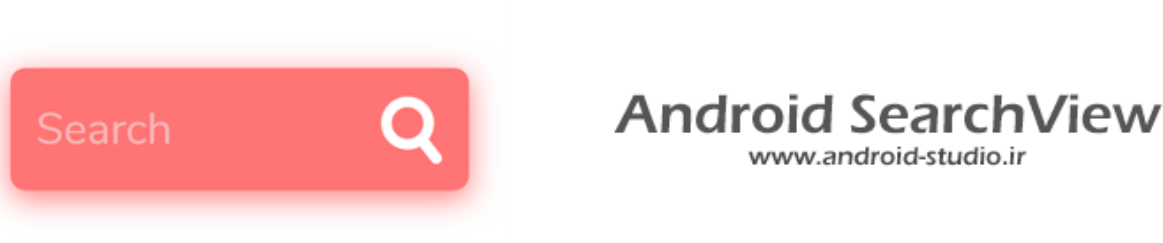
**آموزش کار با SearchView**

مدرس : سیدمهدی مطهری

[www.android-studio.ir](http://www.android-studio.ir)



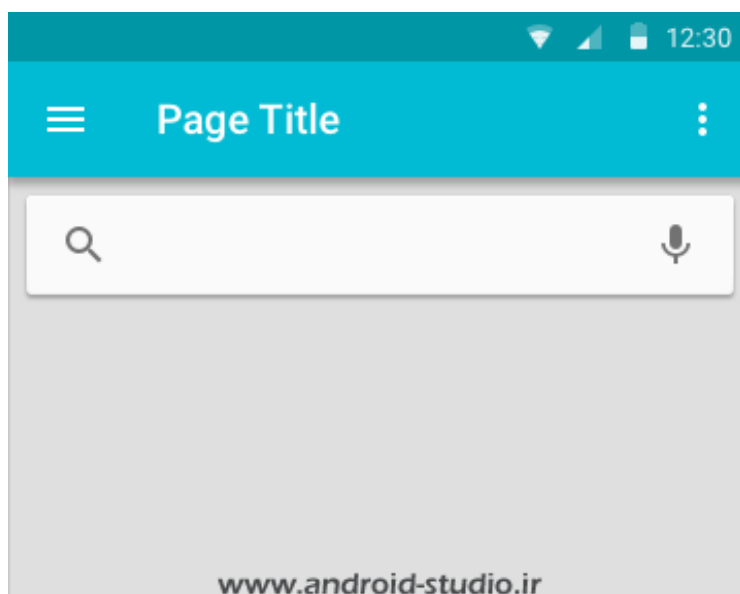
## به نام خدا



تقریباً در بیشتر اپلیکیشن‌ها گزینه‌ای برای جستجو وجود دارد که عمل جستجو بنا به ماهیت و کاربرد برنامه می‌تواند در دیتابیس محلی، سرور و... انجام شده و نتیجه به کاربر برگردانده شود. در این جلسه قصد داریم به صورت ساده نحوه تعریف یک ویجت با نام SearchView را بررسی کنیم.

## SearchView چیست؟

یکی دیگر از ویجت‌های کاربردی اندروید است که امکان جستجوی یک یا چند کاراکتر را برای کاربر فراهم کرده و نتیجه جستجو را در قالب یک لیست نمایش می‌دهد.



ویجت SearchView را می‌توان به تولبار (اکشن بار) هم اضافه کرد. این ویجت لیستی از نتایج (در صورت وجود) را نمایش داده و امکان انتخاب یک گزینه توسط کاربر را هم فراهم می‌سازد. در ادامه جلسه یک SearchView را ابتدا در محیط اصلی اکتیویتی و سپس در Toolbar اضافه می‌کنم.



## پروژه SearchView در اندروید استودیو

ابتدا طبق مبحث آموزش ساخت پروژه در اندروید استودیو یک پروژه اندرویدی با نام SearchView می‌سازم. اکتیویتی را از نوع Empty Activity و زبان را Java انتخاب کردم.

ابتدا در layout اکتیویتی یک ListView و SearchView اضافه می‌کنم. در مبحث آموزش کار با ListView قبلا با این ویجت آشنا شدیم. لازم است قبل از مطالعه ادامه آموزش ابتدا به مبحث ListView مسلط باشید.

activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:divider="#B3B3B3"
        android:dividerHeight="1dp"
        android:layout_below="@id/search_view"/>

    <SearchView
        android:id="@+id/search_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:queryHint="... جستجو"
        android:iconifiedByDefault="false"
        android:layout_alignParentTop="true"/>

</RelativeLayout>
```

دو ویژگی queryHint و iconifiedByDefault به ترتیب برای اضافه کردن راهنما و نحوه نمایش گزینه جستجو استفاده می‌شوند. در اینجا برای ویژگی دوم مقدار false تعریف شده بنابراین ویجت به شکل آیکن تنها نخواهد بود و کادر جستجو هم نمایش داده می‌شود. در صورتی که مقدار true باشد تنها یک آیکن ذره بین نمایش داده می‌شود که با کلیک روی آن، کادر جستجو نمایان می‌شود.

سپس دو ویجت را در اکتیویتی تعریف کرده و یک لیست با چند مقدار می‌سازم که شامل نام چند شهر است:



## MainActivity.java

```
package ir.android_studio.searchview;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.SearchView;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    SearchView searchView;
    ListView listView;
    ArrayList list;
    ArrayAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        searchView = findViewById(R.id.search_view);
        listView = findViewById(R.id.list_view);

        list = new ArrayList<>();
        list.add("تهران");
        list.add("مشهد");
        list.add("فردوس");
        list.add("یزد");
        list.add("تبریز");
        list.add("اهواز");
        list.add("ساری");

        adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
list);
        listView.setAdapter(adapter);
    }
}
```

تا اینجا یک لیست ساخته شد که در نهایت توسط متد `setAdapter()` بر روی `ListView` ست شده است.

در قدم بعد باید یک شنونده (`Listener`) برای `SearchView` بسازیم تا ورودی کاربر را بررسی و نتیجه مناسب را برگرداند. برای اینکار از متد `setOnQueryTextListener()` استفاده می‌کنیم. متد را درون `onCreate()` اکتیویتی تعریف می‌کنم:



```
searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String s) {
        return false;
    }

    @Override
    public boolean onQueryTextChange(String s) {
        return false;
    }
});
```

این متد دو متد `onQueryTextSubmit()` و `onQueryTextChange()` دارد که به طور خودکار توسط اندروید استودیو اضافه شد. هردو از نوع `boolean` هستند یعنی `true` یا `false` برمی گردانند. با توجه به نام هر متد می توان کاربرد آن و اینکه چه زمانی اجرا می شود را حدس زد. `Submit` به معنی "ارسال" و `Change` به معنی "تغییر" است. یعنی `onQueryTextChange()` هنگام تغییر مقدار ورودی در کادر جستجو فراخوانی می شود که شامل اضافه یا حذف شدن یک کاراکتر است. اما `onQueryTextSubmit()` هنگام ارسال مقدار نهایی فراخوانی می شود؛ یعنی زمانی که کاربر دکمه `Go` یا `enter` کیبورد را می زند. دو متد را به این صورت تکمیل می کنم:

```
searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String s) {

        if(list.contains(s)){
            adapter.getFilter().filter(s);
        }else{
            Toast.makeText(MainActivity.this, "موردی یافت نشد",
            Toast.LENGTH_LONG).show();
        }

        return false;
    }

    @Override
    public boolean onQueryTextChange(String s) {

        adapter.getFilter().filter(s);

        return false;
    }
});
```

در هردو متد، کاراکترهای وارد شده توسط کاربر در یک `String` با نام `s` ذخیره می شود.



برای `onQueryTextSubmit` یک شرط تعریف کردم به اینصورت که بررسی می‌کند چنانچه مقدار وارد شده توسط کاربر (s) در list موجود بود، بعد از ارسال توسط کاربر، آنرا در `ListView` نمایش دهد. به عبارت دیگر رشته ذخیره شده در s توسط adapter روی `listView` ست شود:

```
adapter.getFilter().filter(s);
```

`getFilter()` گزینه یا گزینه‌های موجود در list که با مقدار ذخیره شده در s برابر باشد را فیلتر کرده و مابقی آیتم‌های لیست را حذف می‌کند. اما چنانچه ورودی کاربر مشابه با هیچکدام از آیتم‌ها نباشد یک `Toast` نمایش داده خواهد شد که اعلام می‌کند موردی یافت نشد.

اما در `onQueryTextChange` شرطی وجود ندارد زیرا فقط ورودی کاربر را با لیست چک کرده و چنانچه مورد یا موارد یکسانی وجود داشت، فیلتر کرده و در `ListView` نمایش می‌دهد. هر بار که کاربر یک کاراکتر کم یا اضافه کند، لیست بروز خواهد شد.

کد کامل اکتیویتی:

### MainActivity.java

```
package ir.android_studio.searchview;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    SearchView searchView;
    ListView listView;
    ArrayList list;
    ArrayAdapter adapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        searchView = findViewById(R.id.search_view);
        listView = findViewById(R.id.list_view);
        list = new ArrayList<>();
        list.add("تهران");
        list.add("مشهد");
    }
}
```



```
list.add("فردوس");
list.add("یزد");
list.add("تبریز");
list.add("اهواز");
list.add("ساری");
adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
list);
listView.setAdapter(adapter);

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String s) {

        if(list.contains(s)){
            adapter.getFilter().filter(s);
        }else{
            Toast.makeText(MainActivity.this, "موردی یافت نشد",
Toast.LENGTH_LONG).show();
        }

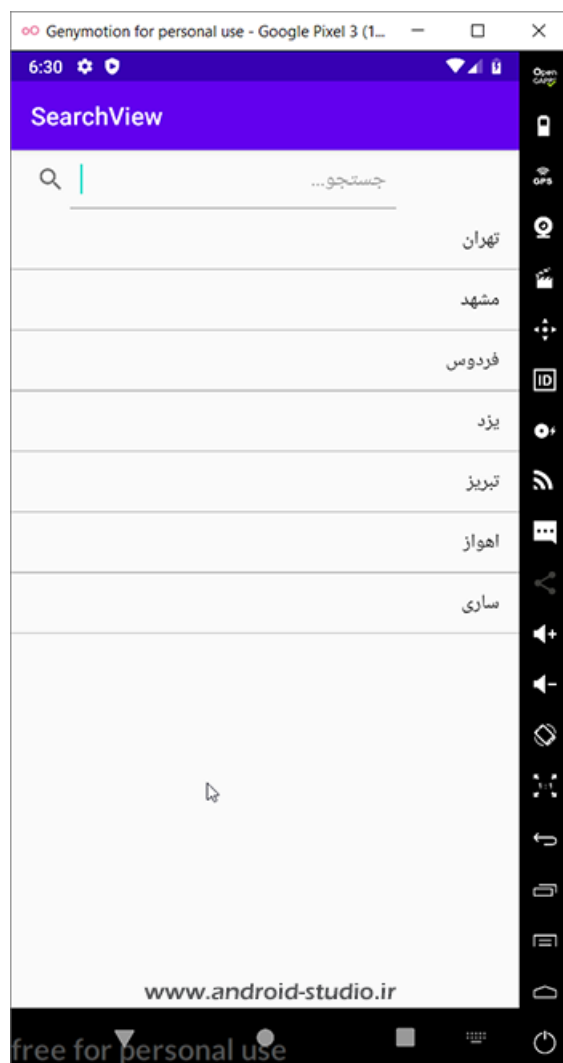
        return false;
    }

    @Override
    public boolean onQueryTextChange(String s) {

        adapter.getFilter().filter(s);

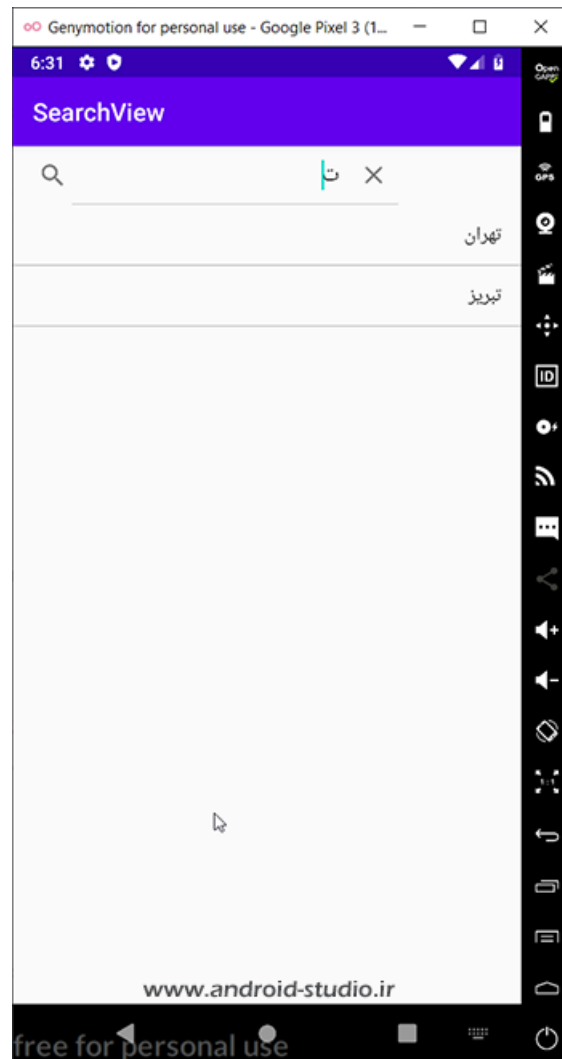
        return false;
    }
});
}
```

پروژه را اجرا می‌کنم:

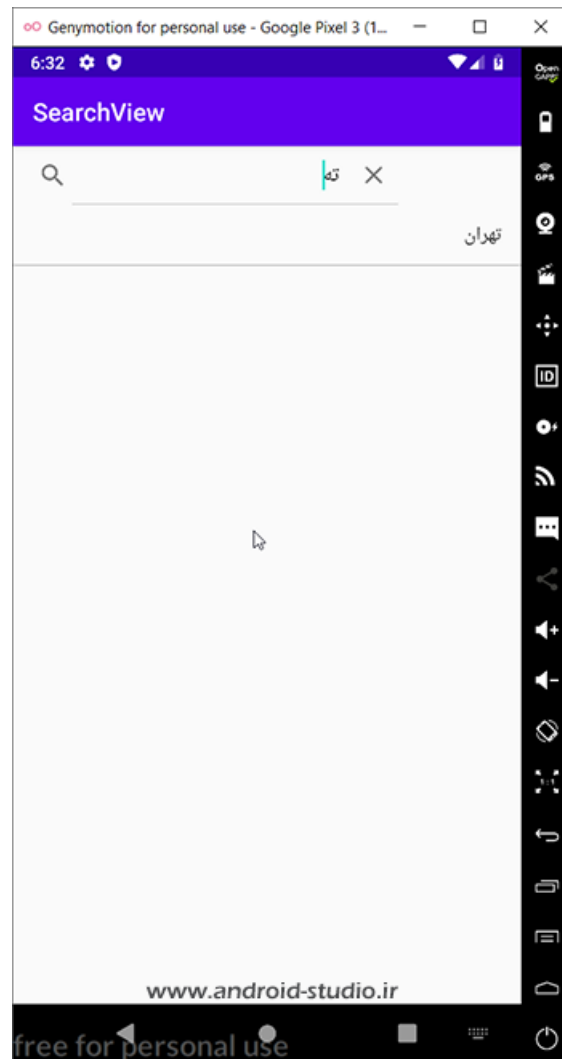


حرف "ت" را در کادر جستجو وارد می‌کنم:



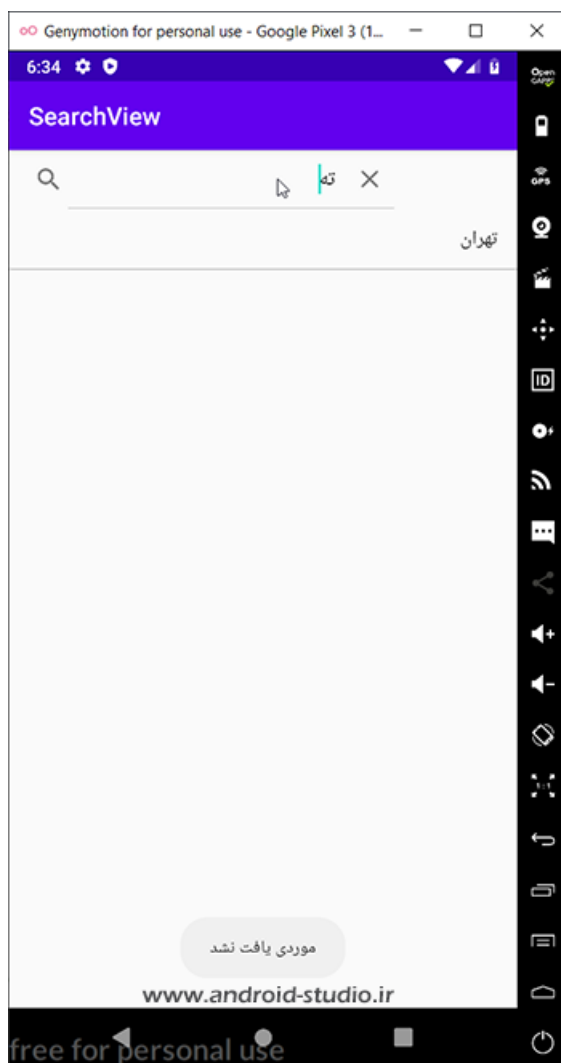


با وارد شدن این کاراکتر، متد `onQueryTextChange` اجرا شده و دو آیتمی که با حرف "ت" آغاز شده به صورت فیلتر شده در `ListView` نمایش داده شده است. حالا حرف "ه" را اضافه می‌کنم:



با تغییر مجدد ورودی و اجرای دوباره‌ی متد، فیلتر مجدد اعمال شد که فقط یک گزینه از دو گزینه قبل در لیست باقی ماند.

دکمه enter یا Go کیبورد را می‌زنم:



متد `onQueryTextSubmit` فراخوانی شده و چون آیتمی با مقدار "ته" در `list` تعریف نشده بنابراین پیغام "موردی یافت نشد" اجرا می‌شود.

## اضافه کردن `SearchView` به `Toolbar` (اکشن بار)

در این قسمت قصد داریم `SearchView` را در `Toolbar` اکتیویتی نمایش دهیم. البته نحوه اضافه کردن آن به `Toolbar` با حالت عادی متفاوت است و باید به عنوان یک آیتم `Menu` تعریف شود که قبلاً در مبحث [آموزش `Toolbar` \(در اندروید\)](#) آشنا شدیم. ابتدا دایرکتوری `menu` را ساخته سپس یک فایل `xml` به آن اضافه می‌کنم. من نام این فایل را `toolbar_searchview` انتخاب کردم:



**New Resource File** [X]

File name:  ↑↓

Resource type:  ▼

Root element:

Source set:  ▼

Directory name:

Available qualifiers:

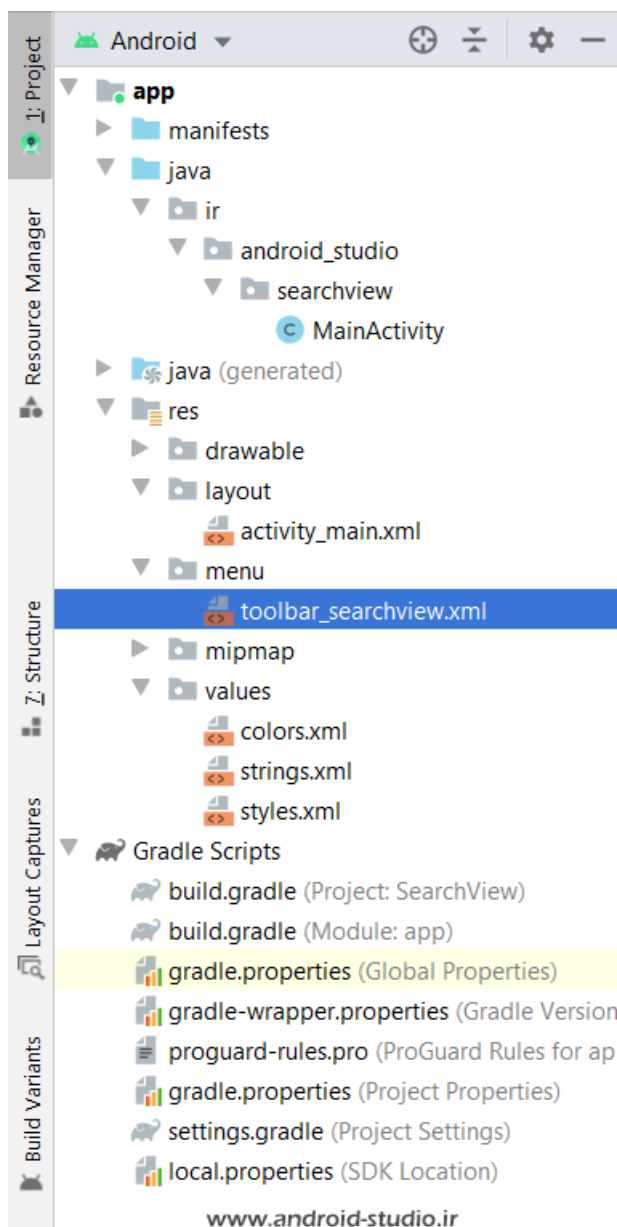
- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode

Chosen qualifiers:

Nothing to show

www.android-studio.ir

OK Cancel Help



در منو یک آیتم به صورت زیر تعریف می‌کنم:



## toolbar\_searchview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/toolbar_search"
        android:icon="@drawable/ic_search"
        android:title="Search"
        app:showAsAction="ifRoom|withText"
        app:actionViewClass="android.widget.SearchView"/>

</menu>
```

یک آیکون ذره بین از آیکون‌های پیش فرض خود اندروید استودیو (Vector Assets) به drawable پروژه اضافه کردم که در اینجا استفاده شده. در توله‌بار بر خلاف قسمت قبل که یک ویجت از نوع SearchView به layout اضافه شد، یک ویژگی با نام actionViewClass و با مقدار android.widget.SearchView تعریف شده که این آیتم را با ظاهر یک کادر جستجو نشان می‌دهد.

در قدم بعد در کلاس اکتیویتی و بعد از متد onCreate() یکی دیگر از متدهای اکتیویتی با نام onCreateOptionsMenu() را اضافه می‌کنم:

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    return super.onCreateOptionsMenu(menu);
}
```



متد را به اینصورت تکمیل می‌کنم:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

    MenuInflater mInflater = getMenuInflater();
    mInflater.inflate(R.menu.toolbar_searchview, menu);
    final SearchView toolbarSearchView =
        (SearchView) menu.findItem(R.id.toolbar_search).getActionView();
    toolbarSearchView.setQueryHint("جستجو");
    toolbarSearchView.setIconifiedByDefault(true);

    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String s) {

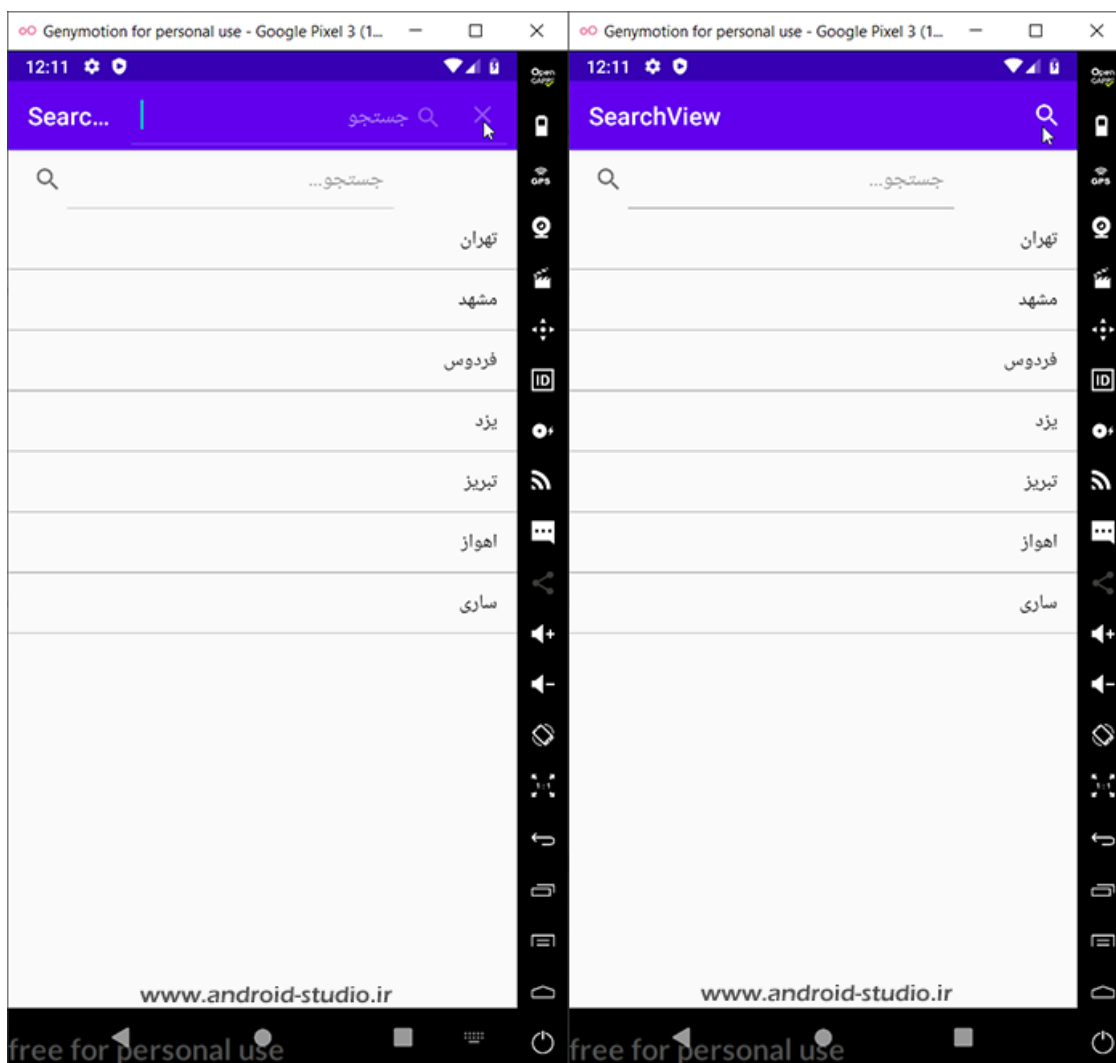
            if(list.contains(s)){
                adapter.getFilter().filter(s);
            }else{
                Toast.makeText(MainActivity.this, "موردی یافت نشد",
                    Toast.LENGTH_LONG).show();
            }

            return false;
        }

        @Override
        public boolean onQueryTextChange(String s) {
            adapter.getFilter().filter(s);
            return false;
        }
    });

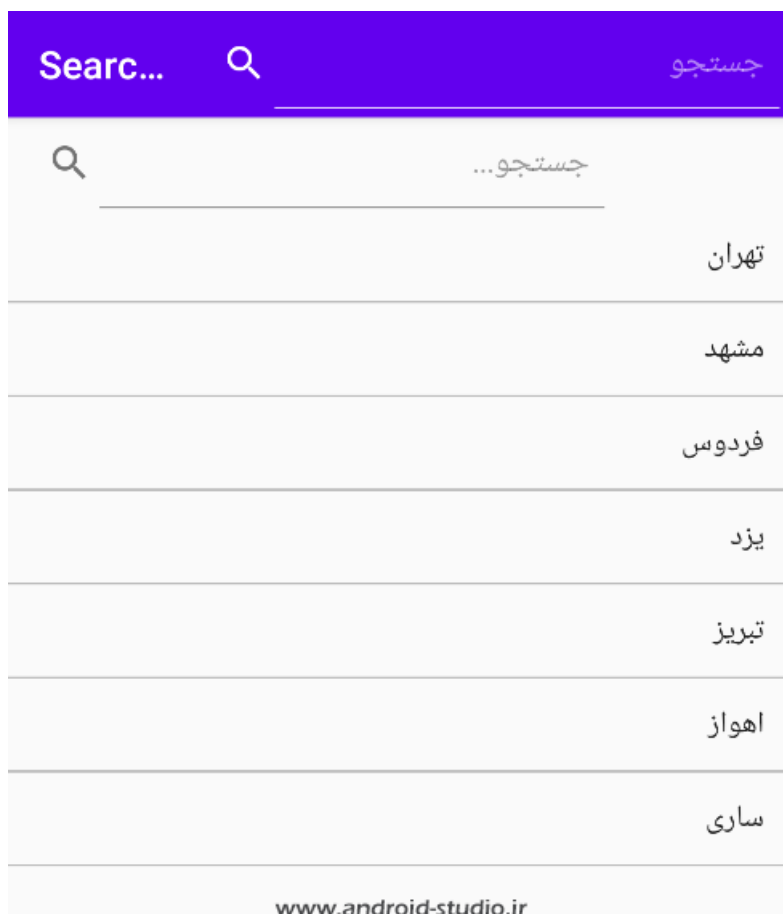
    return super.onCreateOptionsMenu(menu);
}
```

بعد از inflate کردن menu، یک SearchView با نام toolbarSearchView ساختم که به آیتم menu متصل شده است. در ادامه از دو متد setQueryHint و setIconifiedByDefault استفاده شده که اولی یک Hint یا راهنما با عنوان "جستجو" به کادر اضافه کرده و دومی چون مقدار true گرفته در حالت پیش فرض و تا قبل از کلیک روی آیکون جستجو، صرفاً آیکون آن نمایش داده می‌شود و نه کادر جستجو. در نهایت مانند قسمت قبل متد setOnQueryTextListener تعریف شده است. پروژه را اجرا می‌کنم:



مشاهده می‌کنید یک آیکن به سمت راست تولبار اضافه شده که با کلیک روی آن، کادر جستجو باز شده و مانند قسمت قبل امکان جستجو در لیست را داریم. چنانچه برای `setIconifiedByDefault` مقدار `false` قرار بدهیم، کادر جستجو به طور پیش فرض نمایش داده می‌شود:





در این آموزش حالت ساده یک `SearchView` را در اندروید بررسی کردیم. بسته به نوع نیاز ما به این ویجت در پروژه‌های مختلف، پیاده سازی آن از پیچیدگی بیشتری برخوردار خواهد بود. یک نمونه آن، دریافت اطلاعات از سمت سرور و نمایش نتیجه در قالب `RecyclerView` است که در جلسه آموزش کار با نقشه سیدار مپ در اندروید به آن پرداخته شده است.

کد نهایی اکتیویتی:

`MainActivity.java`

```
package ir.android_studio.searchview;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.SearchView;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
```



```

SearchView searchView;
ListView listView;
ArrayList list;
ArrayAdapter adapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    searchView = findViewById(R.id.search_view);
    listView = findViewById(R.id.list_view);

    list = new ArrayList<>();
    list.add("تهران");
    list.add("مشهد");
    list.add("فردوس");
    list.add("یزد");
    list.add("تبریز");
    list.add("اهواز");
    list.add("ساری");
    adapter = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1,
list);
    listView.setAdapter(adapter);

    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(String s) {

            if(list.contains(s)){
                adapter.getFilter().filter(s);
            }else{
                Toast.makeText(MainActivity.this, "موردی یافت نشد",
Toast.LENGTH_LONG).show();
            }

            return false;
        }

        @Override
        public boolean onQueryTextChange(String s) {

            adapter.getFilter().filter(s);

            return false;
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {

    MenuInflater mInflater = getMenuInflater();
    mInflater.inflate(R.menu.toolbar_searchview, menu);
    final SearchView toolbarSearchView = (SearchView)
menu.findItem(R.id.toolbar_search).getActionView();

```



```

toolbarSearchView.setQueryHint("جستجو");
toolbarSearchView.setIconifiedByDefault(true);

toolbarSearchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String s) {

        if(list.contains(s)){
            adapter.getFilter().filter(s);
        }else{
            Toast.makeText(MainActivity.this, "موردی یافت نشد",
Toast.LENGTH_LONG).show();
        }

        return false;
    }

    @Override
    public boolean onQueryTextChange(String s) {
        adapter.getFilter().filter(s);
        return false;
    }
});

return super.onCreateOptionsMenu(menu);
}
}

```

موفق و پیروز باشید.

**مطالعه بیشتر:**

<https://developer.android.com/reference/android/widget/SearchView>

<https://developer.android.com/guide/topics/search/search-dialog>

<https://developer.android.com/reference/android/widget/SearchView.OnQueryTextListener>

**توجه: سورس پروژه درون پوشه Exercises قرار دارد**

**با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش‌های بهتر یاری فرمائید.  
این فایل رایگان بوده و انتشار آن (بدون دخل و تصرف) مانعی ندارد.**

[www.android-studio.ir](http://www.android-studio.ir)