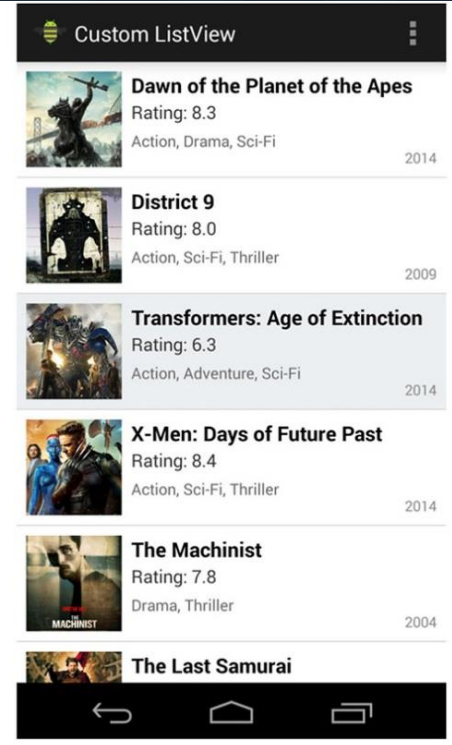
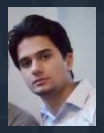


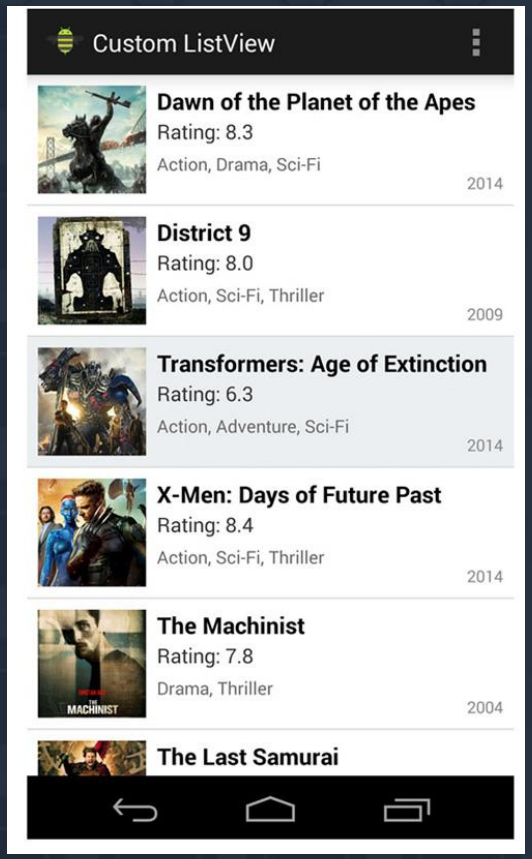
Custom Listview

لیست سفارشی

احسان قربان نژاد
فناوری اطلاعات پارمیس
info@ehsanet.ir



آنچه میخواهیم انجام دهیم...



activity_main.xml

```
<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"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:divider="@color/list_divider"
        android:dividerHeight="1dp"
        android:listSelector="@drawable/list_row_selector" />

</RelativeLayout>
```

مروری بر یک فایل **selector** برای هایلایت کردت آیتم های لیست در هنگام انتخاب

list_row_selector.xml

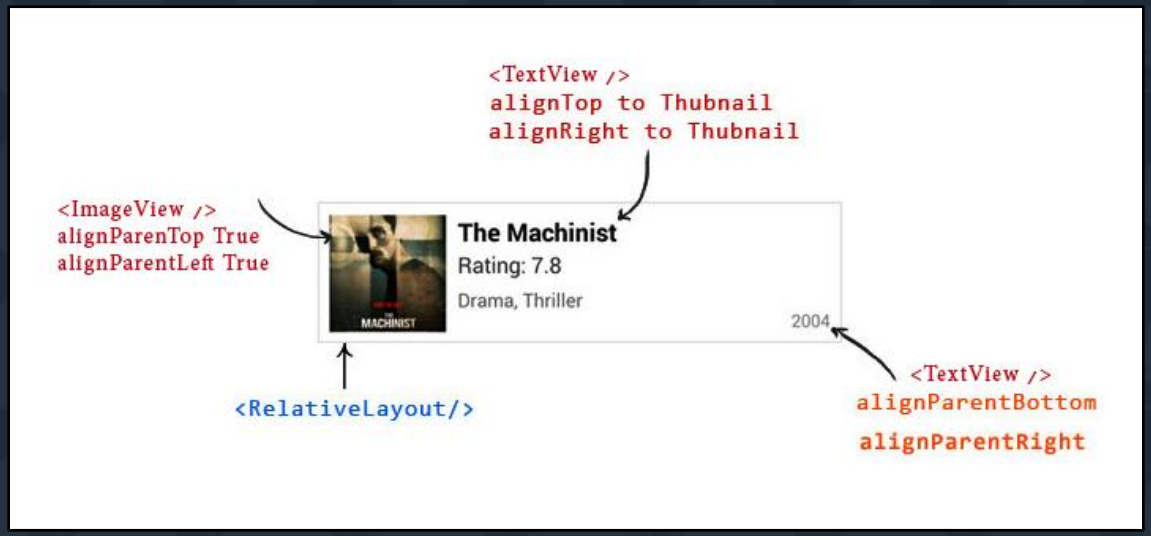
```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/list_row_bg" />
    <item android:drawable="@drawable/list_row_bg_hover" android:state_pressed="true"/>
    <item android:drawable="@drawable/list_row_bg_hover" android:state_pressed="false"
                                                         android:state_selected="true"/>

</selector>
```

قالب هر آیتم لیست

- یک فایل xml تعریف میکنیم.

```
<RelativeLayout >
   <ImageView android:id="@+id/thumbnail" />
   <TextView android:id="@+id/title" />
   <TextView android:id="@+id/rating"/>
   <TextView android:id="@+id/genre" />
   <TextView   android:id="@+id/releaseYear"   />
</RelativeLayout>
```



نحوه تعریف و مقدار دهی در Activity

```
public class MainActivity extends Activity {

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

            List<Movie> movieList = createMovieList();
            ListView  = (ListView) findViewById(R.id.list);
            CustomListAdapter adapter = new CustomListAdapter(this, movieList);
            listView.setAdapter(adapter);
    }
}
```

کلاس CustomListAdapter

- سازنده کلاس (Constructor)

```
public class CustomListAdapter extends BaseAdapter {
    private final Context context;
    private final LayoutInflater  inflater;
    private List<Movie> items;

    public CustomListAdapter(Context c , List<Movie> movies){
        this.context = c;
        this.items = movies;
        this.inflater =(LayoutInflater)  context.getSystemService( Context.LAYOUT_INFLATER_SERVICE ) );
    }
}
```

نکته: LayoutInflater یک سرویس سیستمی برای ساختن(پر کردن) یک شی از نوع View از روی فایل xml متناظرش.

نکته: BaseAdapter یک کلاس پایه که امکان ساخت یک آداپتر سفارشی از طریق ارث بردن از آن و پیاده سازی بعضی از متد هایش، را می دهد.

پیاده سازی های CustomListAdapter

```
@Override
public int getCount() {
        return _items == null ? 0 : items.size();
}
@Override
public Object getItem(int position) {
        return items == null ? null : items.get(position);
}

@Override
public long getItemId(int position) {
        return items == null ? 0 : items.get(position).getId();
}
```

نکته: Position ای که اندروید در این متدها در اختیار ما قرار میدهد، موقعیت عنصر درون لیست است. مثلا عنصر صفرم، یکم و یا غیره. ما با داشتن این شماره موقعیت میتوانیم به عناصر لیست که در این کلاس با نام Items تعریف شده اند، دسترسی داشته باشیم (از طریق متد getبر روی لیست)

پیاده سازی های CustomListAdapter

- پیاده سازی متد چهارم getView ()

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    if (convertView == null)
        convertView = inflater.inflate(R.layout.list_row, null);

    // now we can find our item_layout controls  - for example:
    ImageView thumb = (ImageView) convertView.findViewById(R.id.thumbnail);
    TextView title = (TextView) convertView.findViewById(R.id.title);

   // whatever we want to bind to  our controls  - for example :
    title.setText(items.get(position).getTitle();

    return convertView;
}
```

نکته: convertView چیست؟

فرض کنید ۱۵ آیتم برای نمایش دارید. صفحه موبایل شما در هر لحظه ۵ تا را میتواند نمایش دهد. مابقی نیاز به اسکرول کردن دارند.

در متد getView تنها ۵ convertView ساخته میشود. برای مابقی لیست از همین ۵ تا استفاده میشود به اینصورت که اطلاعات درون آن تنها عوض می شود.

لذا شرط null بودن برای عنصر ششم دیگر False برمی گرداند و نیاز به ساخت مجدد نداریم.

GetViewدر Performanceبهبود

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
      ViewHolder holder;
if (convertView == null)
      convertView = inflater.inflate(R.layout.list_row, null);
      holder = new ViewHolder ();
      holder.title = (TextView) convertView.findViewById(R.id.title);
      convertView.setTag(holder);
 } else
      holder = (ViewHolder ) convertView.getTag();
 hloder.title.setText(item.get(position).getTitle();
 return convertView;
}
```

نکته:

دستور setTag برای نگهداری یک holder در convertView متناظر با خودش استفاده شده است. دستور setTag متد عامی است که هرجا خواستیم چیزی را(ازهرنوع دیتاتایپی) در view مان نگهداریم استفاده میکنیم.

* برای آشنایی با ViewHolder اسلاید بعدی را مشاهده نمایید.

بهبود Performance در GetView

- کلاس ViewHolder

```
static class ViewHolder {

    TextView title;
    TextView genre;
    TextView releaseYear;
    TextView rating;
    ImageView thumb;
}
```

قالب همه ی آیتم ها یکسان است. اگر طبق مثال قبل، ما بجای ۱۵ convertView، ۵تا میسازیم، چرا باید در هر ۱۵ بار کنترل های آن آیتم را find کنیم؟

بجای اینکار فقط برای همان ۵ تا این کار را انجام می دهیم. برای اینکار یک کلاس سفارشی و static به نام ViewHolder میسازیم.
تعریف کلاس ViewHolderبصورت داخلی تعریف میشود(inner class) یعنی در داخل بدنه کلاس CustomLisAdapter و البته خارج از متد getView !

انیمیشن در ListView

- در متد GetView میتوان انیمیشن را بر روی convertView اجرا کرد.

```
//inside getView
Animation animation =
        AnimationUtils.loadAnimation(context, (position > lastPosition )
        ? R.anim.up_from_bottom : R.anim.down_from_top);
v.startAnimation(animation);
lastPosition = pos;
```

نکته:

private int lastPosition = -1; را در ابتدای کلاس CustomListAdapter اضافه کنید.

نکته:

up_from_bottom و down_from_top نام دو فایل انیمیشن با پسوند xmlآست که در پوشه anim داخل پوشه res باید قرار گیرد. یک نمونه از یک انیمیشن:

```
<?xml version="1.0" encoding="utf-8"?>
<translate
android:fromXDelta="90%" android:toXDelta="0%"
android:fromYDelta="0%" android:toYDelta="0%"
android:duration="400" />
```