

Android ListView Item'ları Dinamik Olarak Düzenlemek

Android uygulamasında **ListView'in** item'larını **dilediğimiz gibi düzenlemek istiyorsak özel bir Adapter sınıfı oluşturmamız** gerekmektedir. Adapter sınıfını Activity sınıfı içerisinde **inner sınıf olarak** tanımlayabiliriz. Bu işlemlerin nasıl gerçekleştiğini açıklamak için örnek bir uygulama yapalım.

Uygulamamızda **ListView** elementine custom bir layout ekleyeceğiz. Bu layout içerisinde hem **TextView** elementi hem de **ImageView** elementini kullanacağız. Uygulamanın amacı, belirli bir sonuca göre **ImageView** elementini **göstermek/gizlemek** ve **TextView'in** text'ini **renkli** yazdırmaktır.

Gereken Adımlar

1. **ListView** Layout
2. Custom **ListView** Item Layout
3. Custom **ListView** Item Sınıfı
4. Custom **ArrayAdapter**
5. Activity Sınıfı

ListView Layout (main.xml)

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android (http://schemas.android.com
2 /apk/res/android) "
3     android:orientation="vertical"
4     android:layout_width="fill_parent"
5     android:layout_height="fill_parent"
6     >
7     <ListView
8         android:layout_width="match_parent"
9         android:layout_height="0dp"
10        android:layout_weight="1"
11        android:divider="#ff80c6ee"
12        android:dividerHeight="0.3dp"
13        android:fadingEdgeLength="0dp"
14        android:id="@+id/listWords"
15        />
</LinearLayout>
```

Custom ListView Item Layout (view_list_item.xml)

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android (http://schemas.android.com
2 /apk/res/android) "
3     android:layout_height="match_parent"
4     android:layout_width="wrap_content"
5     android:orientation="horizontal">
6
7     <ImageView
8         android:id="@+id/ac_list_view_frequent"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:paddingTop="5dip"
12        android:paddingRight="5dip"
13        android:src="@drawable/red_key_icon"
14        android:contentDescription="@string/ac_home_img_description_verbs"/>
15
16     <TextView
17         android:id="@+id/ac_list_view_txt_content"
18         android:layout_weight="1"
19         android:textSize="20sp"
20         android:paddingTop="5dip"
21         android:paddingBottom="5dip"
22         android:layout_width="0dp"
23         android:layout_height="wrap_content"
24         android:textColor="#ff000000"/>
25
</LinearLayout>
```

Custom ListView Item Sınıfı

```
1 public class WordIdName implements Serializable {
2     private long id;
3     private String name;
4     private WordType type;
```

```

5     private boolean frequent;
6     public long getId() {
7         return id;
8     }
9
10    public void setId(long id) {
11        this.id = id;
12    }
13
14    public String getName() {
15        return name;
16    }
17
18    public void setName(String name) {
19        this.name = name;
20    }
21
22    public boolean isFrequent() {
23        return frequent;
24    }
25
26    public void setFrequent(boolean frequent) {
27        this.frequent = frequent;
28    }
29
30    @Override
31    public String toString() {
32        return name;
33    }
34 }

```

Custom ArrayAdapter Sınıfı

```

1     private class MyAdapter extends ArrayAdapter<WordIdName> {
2
3         public MyAdapter(Context context, int resource, int textViewResourceId, List<WordIdName> wordIdNames)
4     {
5         super(context, resource, textViewResourceId, wordIdNames);
6     }
7
8     /**
9     * ListView elementi icerisinde ne kadar item varsa o kadar execute edilir
10    * Ornegin 1000 tane item varsa 1000 kez bu metod calistirilir.
11    * Ayrinti icin
12    * <a href="http://stackoverflow.com/questions/12400338/explanation-of-the-getview-method-of-an-
13    arrayadapter (http://stackoverflow.com/questions/12400338/explanation-of-the-getview-method-of-an-
14    arrayadapter)">
15    * tiklayiniz</a>
16    * @param position 0 degerinden baslayarak item sayisi kadar artar. Itemlerin index'ini temsil eder
17    * @param convertView
18    * @param parent
19    * @return
20    */
21    @Override
22    public View getView(int position, View convertView, ViewGroup parent) {
23        //ListView elementinin position degerine gore item'ini getirir
24        WordIdName wordIdName = super.getItem(position);
25        //view nesnesi icerisinde hem ImageView hem de TextView barindirir. Cunku view_list_item layout
26        View view = super.getView(position, convertView, parent);
27        ImageView frequent = (ImageView) view.findViewById(R.id.ac_list_view_frequent);
28        TextView content = (TextView) view.findViewById(R.id.ac_list_view_txt_content);
29        if (!wordIdName.isFrequent()) {
30            frequent.setVisibility(View.GONE);
31            content.setTextColor(Color.RED);
32        }
33        return view;
34    }
35 }

```

Activity Sınıfı

```

1     public class MainActivity extends Activity {
2         @Override
3         protected void onCreate(Bundle savedInstanceState) {
4             super.onCreate(savedInstanceState);
5             setContentView(R.layout.main);
6
7             ListView listWords = (ListView) findViewById(R.id.listWords);

```

```
8     ArrayAdapter<WordIdName> adapter = new MyAdapter(this,
9         R.layout.view_list_item, R.id.ac_list_view_txt_content, getAllWordNames());
10    registerForContextMenu(listWords);
11    listWords.setAdapter(adapter);
12    }
13    public List<WordIdName> getAllWordNames() {
14        //////////////////////////////////
15    }
16 }
```

Not: ArrayAdapter<WordIdName> şeklinde MainActivity sınıfında adapter nesnesi yaratıldığı için ListView nesnesi WordIdName item'larını listelemek için, bu sınıfın toString() metodunu çağırır. Dikkat ederseniz WordIdName sınıfında biz bu metodu override ettik.