

# List View

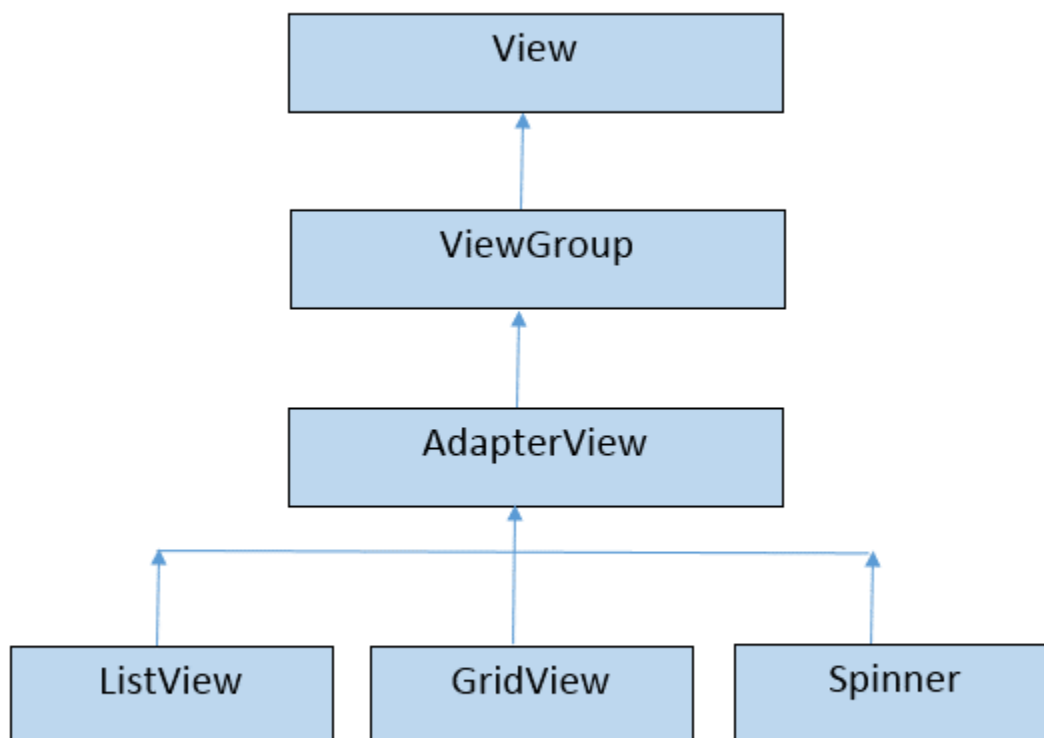
---

## Uvod

Prikaz kolekcije podataka u Androidu urađeno je upotrebom *AdapterView* klase i njenih podklasa.

Podklase koje se koriste su:

- *GridView* – Komponenta za prikaz kolekcije kao mrežu podataka
- *Spinner* – Komponenta za prikaz koja korisniku dozvoljava da iz liste izabere jedan objekat
- *ListView* – Komponenta za prikaz kolekcije gde su podaci postavljeni jedan ispod drugog



Slika 1 AdapterView hijerarhija

## ListView

*ListView* je komponenta koja nam služi za prikaz liste objekata. Ova komponenta ne poznaje detalje objekata koje sadrži, već ih traži od *Adaptera* čiji je zadatak da objekte pripremi za *ListView*.

```
<ListView
    android:id="@+id/list_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```



*Slika 2 ListView*

Reakcija na dodir na jedan od elemenata liste implementirana je kroz *ListView.OnItemClickListener* interfejs klasu i kroz metodu te klase *OnItemClick*.

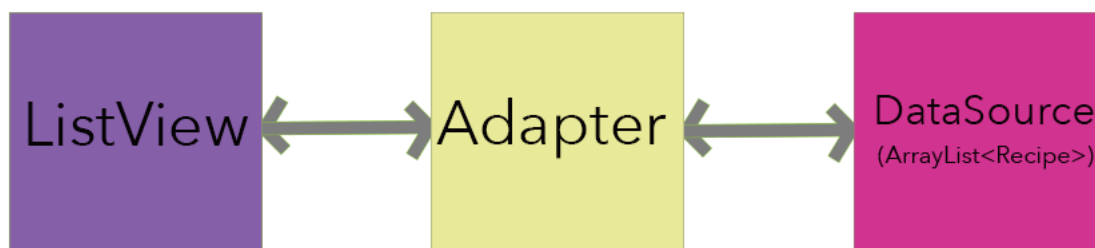
```
mListView.setOnItemClickListener(new OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id)
    {
        /* Your code here */
    }
});
```

## Adapter

*Adapter* klasa predstavlja spregu izmedju podataka i *ListView* komponente. Njegov zadatak je da pruži model podataka za listu i da konvertuje podatke u polja liste. Android predstavlja dva standardna *Adapter*-a:

- *Array Adapter* – Radi sa izvorom podataka u obliku niza ili liste
- *Cursor Adapter* – Radi sa bazom kao izvorom podataka

Android definiše osnovne *layout*-e za prikaz jednostavnijih podataka. Omogućeno je korišćenje sopstvenog *layout*-a za izgled jednog polja *ListView* komponente.



*Slika 3 Adapter*

Android omogućava i definisanje Custom *Adapter*-a korišćenjem klase *BaseAdapter*. To znači da je moguće potpuno prilagoditi *ListView* našim potrebama.

```
private class MyAdapter extends BaseAdapter {  
    // override other abstract methods here  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup container)  
    {  
        if (convertView == null) {  
            convertView = getLayoutInflater().inflate(R.layout.list_item,  
container, false);  
        }  
  
        ((TextView) convertView.findViewById(android.R.id.text1))  
            .setText(getItem(position));  
        return convertView;  
    }  
}
```

*BaseAdapter* je interfejs koji implementira *getView* metodu. Ova metoda dobavlja *View* objekat zadužen za prikaz podataka. *ListView* komponenta implementira Recycle mehanizam. Ovaj mehanizam ima za zadatak da ne kreira uvek novi *View* objekat za svaki podatak već da ponovo upotrebi već kreirane. Na taj način, u slučaju velikog broja podataka, postojaće samo onaj broj *View* objekta koji je moguće prikazati na ekranu. Pri spustanje liste na dole, ili podizanju liste na gore, *View* objekti se neće unistavati, već će se menjati samo podaci koji oni prikazuju. Na ovaj način se štedi memorija za prikaz podataka. Parametar *convertView* metode *getView* je parametar koji sadrži *View* objekat koji bi trebalo kreirati ili menjati.

## ViewHolder

ViewHolder je dizajn patern koji omogućava pristupanje *View* objektu svakog elementa bez bespotrebnog pretražavanja *View* upotrebom *findViewById* metode. Na ovaj način se dobija bolji performans *ListView* komponente. Performans se može dodatno poboljšati korišćenjem druge niti kako bi se glavna nit oslobodila većeg dela obrade.

```
static class ViewHolderItem {
    TextView textViewItem;
}

/* MyAdapter class */

public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolderItem viewHolder;

    if (convertView == null) {
        convertView = getLayoutInflater().inflate(R.layout.list_item, container, false);

        viewHolder = new ViewHolderItem();
        viewHolder.textViewItem = (TextView) convertView
            .findViewById(R.id.textViewItem);

        convertView.setTag(viewHolder);
    } else {
        viewHolder = (ViewHolderItem) convertView.getTag();
    }

    /* Get data Object on position from list/database */

    viewHolder.textViewItem.setText(getItem(position));

    return convertView;
}
```