ListView en Android

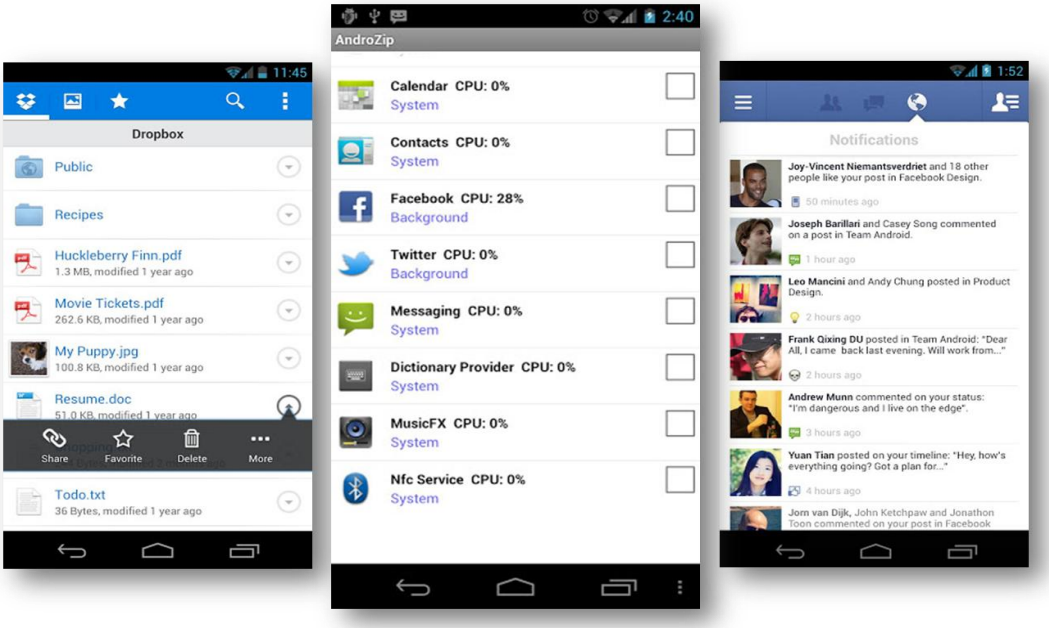
Juan Antonio Jápon de la Torre

09/09/2016

Contenido

ListView en Android

Son las llamadas listas que mostraran al usuario de forma ordenada y organizada una serie de datos:



Tipos de ListView

En Android cuando estamos desarrollando podemos encontrarnos 2 tipos:

- Los que vienen ya diseñados y aporta android para poder usarlos por defecto
- ListView personalizados creados por nosotros

Creación de un ListView no Personalizados

Lo primero que tenemos que hacer es añadirlo o crearlo en nuestra vista. Para ello añadimos el componente y le ponemos un id por defecto.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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="com.android.listviewejemplo.MainActivity"
    android:weightSum="1">
    //Nuestro ListView
    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/listaPersonas"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_weight="1.45" />
</LinearLayout>
```

Una vez añadido nos desplazamos hacia nuestra actividad (MainActivity.class). En ella tendremos que realizar los siguientes pasos:

- **Creamos las variables globales que necesitaremos**

```
String[] personas;
ListView lista;
```

- **Inicializamos el ListView**

```
lista = (ListView) findViewById(R.id.listaPersonas);
```

- **Insertamos datos en nuestro Array:**

```
personas = new String[]
{"Pepe","Miguel","María","Ana","Paco","Japón","Antonio"};
```

- **Creamos un Adaptador:**

```
ArrayAdapter<String> adaptador = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, personas);
```

  Los parámetros a pasarle son:
  - **Contexto**: si estamos trabajando en un Activity el contexto es (**this** o **NombreClaseActivity**.this). Si estamos trabajando con fragmento tendríamos que llamar al contexto con **getActivity()**
  - **Nombre del tipo de layout para mostrar la lista:** En Android existen layout predefinidos que pueden ser usados llamándolos con (android.R.layout.**nombrelayout**)

    Ejemplo: `android.R.layout.simple_list_item_1`
  - **Lista a mostrar**

- **Añadimos el adaptador a nuestro ListView**

```
lista.setAdapter(adaptador);
```

- **Para que los ítems de la lista tenga Eventos implementamos el método setOnItemClickListener(). Nos pedirá que implementemos dicho método**

```
lista.setOnItemClickListener(this);

@Override
public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
    //Operaciones a realizar
}
```

Código Final

```
public class MainActivity extends AppCompatActivity implements
AdapterView.OnItemClickListener {

    String[] personas;
    ListView lista;

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

        //Inicializamos nuestro listView
        lista = (ListView) findViewById(R.id.listaPersonas);

        //Creamos una lista de prueba para mostrarla
        personas = new String[]
{"Pepe","Miguel","María","Ana","Paco","Japón","Antonio"};

        //Creamos un adaptador
        ArrayAdapter<String> adaptador = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, personas);

        //Añadimos el adaptador a nuestro listView para que se muestre
        lista.setAdapter(adaptador);

        //Para tener eventos implementamos el metodo setOnClickListener
        lista.setOnItemClickListener(this);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
        Toast.makeText(MainActivity.this, personas[position],
Toast.LENGTH_SHORT).show();
    }
}
```

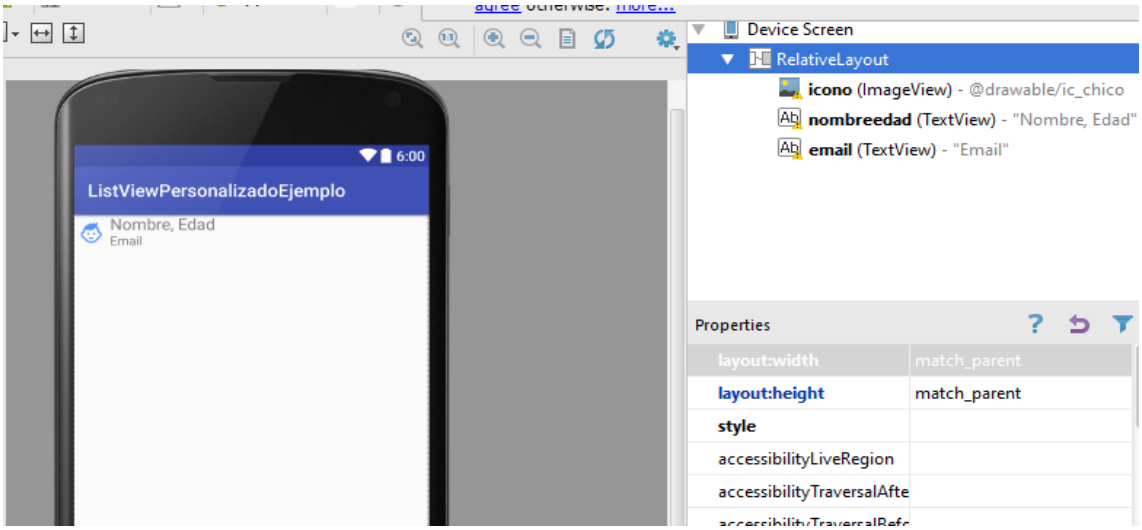
Creación de ListView Personalizados

Al ser personalizados tenemos que realizar las siguientes cosas:

- **Un layout personalizado:** que contendrá la estructura en la que se iran mostrando los ítems del ListView
- **Una clase(Entidad):** Con su constructor y sus getter y setter será la clase que contenga los datos de cada fila (variables a mostrar en la vista)
- **Un Adaptador personalizado:** Será una clase java que extienda de ArrayAdapter <Entidad> ya que mostrara un array de datos en las filas de la vista personalizada ya que cada fila tiene más de 1 dato a diferencia del adaptador por defecto
- **Controlador principal que es el que creara un adaptador del tipo de arriba y lo rellenará de datos**

Layout Personalizado

Se llama en este ejemplo "list_item_personas.xml" y la estructura y los ids de los elementos del layout de este ejemplo será el siguiente:




```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/icono"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:src="@drawable/ic_chico" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="Nombre, Edad"
        android:id="@+id/nombreedad"
        android:layout_alignParentTop="true"
        android:layout_toRightOf="@+id/icono"
        android:layout_toEndOf="@+id/icono" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        android:id="@+id/email"
        android:layout_below="@+id/nombreedad"
        android:layout_toRightOf="@+id/icono"
        android:layout_toEndOf="@+id/icono" />
</RelativeLayout>
```

Una clase (Entidad)

Cada fila tendrá una serie de datos. En este ejemplo estamos mostrando 4 datos:

- El sexo: en forma de icono en función si es hombre "H" o mujer "M"
- El nombre

- La Edad
- El email

Para ello una vez que creamos las variables dentro de la clase debemos crear tanto el constructor como sus getters y setter. Para ello podemos importarlos desde la barra de herramientas dándole a **Code/Generate** Nuestra entidad quedaría de la siguiente manera:

```
public class Persona {
    //Datos a mostrar
    private String nombre;
    private String email;
    private String sexo;

    //Constructores
    public Persona() {
    }

    public Persona(String nombre, String email, String sexo) {
        this.nombre = nombre;
        this.email = email;
        this.sexo = sexo;
    }
    //Getters & Setters
    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }
    public String getEmail() { return email;}
    public void setEmail(String email) { this.email = email; }
    public String getSexo() { return sexo; }
    public void setSexo(String sexo) { this.sexo = sexo; }
}
```

Adaptador Personalizado

Esta sera una clase que extiende de ArrayAdapter y que hara referencia a nuestra entidad creada con anterioridad ya que se encargara de mostrar un array de objetos de esa entidad:

```
public class PersonaAdapter extends ArrayAdapter <NombreEntidad>
```

Una vez creada la clase y crearemos 3 variables para poder trabajar con ellas dentro el adaptador: contexto, ArrayList<NombreEntidad> y idLayoutItem quedando asi:

```
public class NombreEntidadAdapter extends ArrayAdapter <NombreEntidad>{
    Context c;
    ArrayList<NombreEntidad> personas;
    int layoutp;
```

En mi ejemplo

```
public class PersonaAdapter extends ArrayAdapter <Persona>{
    Context c;
    ArrayList<Persona> personas;
    int layoutp;
```

Lo siguiente será crear el Constructor de este adaptador. Podemos importarlo desde **Code/Generate/Constructor** Seleccionando el siguiente:

```
public PersonaAdapter(Context context, int resource, List<Object> objects)
```

Y lo dejaremos completo asi en mi ejemplo con Personas:

```
public PersonaAdapter(Context context, int resource, ArrayList<Persona>
objects) {
    super(context, resource, objects);
    this.c = context;
    this.personas = objects;
    this.layoutp = resource;
}
```

Y por ultimo como todo método que extiende de una clase debemos implementar un método que es el **getView**. Podemos importarlo desde **code/Override Methods** y buscándolo:

```
@Override
public View getView(int position, View convertView, ViewGroup parent){
    //return super.getView(position, convertView, parent);
```

Como se ha realizado comentamos el **return** para que no se use y en su lugar hacemos lo siguiente:

```
//Para recuperar la vista creamos un inflater y a partir de idLayoutItem
recuperamos el layout ya que estamos en un Adaptador
LayoutInflater inflater = (LayoutInflater)
c.getSystemService(c.LAYOUT_INFLATER_SERVICE);
View v = inflater.inflate(layoutp,parent, false);

//Con el objeto View recuperado podemos invocar a findViewById y recuperar los
campos
ImageView icono = (ImageView) v.findViewById(R.id.icono);
TextView nombre = (TextView) v.findViewById(R.id.nombre);
TextView email = (TextView) v.findViewById(R.id.email);

//Recuperamos la persona que se pintará en ese momento en la lista
Persona p = personas.get(position);

//Usando los datos de la Persona rellenamos los campos con sus datos
nombre.setText(p.getNombre());
email.setText(p.getEmail());
if(p.getSexo().equals("H")){
    icono.setImageResource(R.drawable.ic_chico);
}else{
    icono.setImageResource(R.drawable.ic_chica);
}

//devolvemos la vista
return v;
```

De esta manera el código final del Adaptador es el siguiente:

```
public class PersonaAdapter extends ArrayAdapter <Persona>{
    Context c;
    ArrayList<Persona> personas;
    int layoutp;

    public PersonaAdapter(Context context, int resource, ArrayList<Persona>
objects) {
        super(context, resource, objects);
        this.c = context;
        this.personas = objects;
        this.layoutp = resource;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        //return super.getView(position, convertView, parent);

        //Para recuperar la vista creamos un inflater y a partir de
idLayoutItem recuperamos el layout
        LayoutInflater inflater = (LayoutInflater)
c.getSystemService(c.LAYOUT_INFLATER_SERVICE);
        View v = inflater.inflate(layoutp,parent, false);

        //Con el objeto View recuperado podemos invocar a findViewById y
recuperar los campos
        ImageView icono = (ImageView) v.findViewById(R.id.icono);
        TextView nombre = (TextView) v.findViewById(R.id.nombre);
        TextView email = (TextView) v.findViewById(R.id.email);

        //Recuperamos la persona que se pintará en ese momento en la lista
        Persona p = personas.get(position);

        //Usando los datos de la Persona rellenamos los campos con sus datos
        nombre.setText(p.getNombre());
        email.setText(p.getEmail());
        if(p.getSexo().equals("H")){
            icono.setImageResource(R.drawable.ic_chico);
        }else{
            icono.setImageResource(R.drawable.ic_chica);
        }

        //devolvemos la vista
        return v;
    }
}
```

Cargando los datos en el Controlador Principal

Por último solo nos queda cargar datos en el método **onCreate** de nuestro controlador o MainActivity o claseActivity secundaria pará ello haremos lo siguiente

```
//Recuperamos el ListView
lista = (ListView) findViewById(R.id.listView);

//Creamos un array de datos tipo Persona
ArrayList<Persona> personas = new ArrayList<Persona>();

//Rellenamos el Array
personas.add(new Persona("Juan Antonio Japon", "jajapon@gmail.com","H"));
personas.add(new Persona("Maria de las Mercedes","mmercedes@gmail.com","M"));
personas.add(new Persona("Juan Antonio Japon", "jajapon@gmail.com","H"));
personas.add(new Persona("Maria de las Mercedes","mmercedes@gmail.com","M"));
personas.add(new Persona("Juan Antonio Japon", "jajapon@gmail.com","H"));
```

```
personas.add(new Persona("Maria de las Mercedes","mmercedes@gmail.com","M"));
personas.add(new Persona("Juan Antonio Japon", "jajapon@gmail.com","H"));
personas.add(new Persona("Maria de las Mercedes","mmercedes@gmail.com","M"));
personas.add(new Persona("Juan Antonio Japon", "jajapon@gmail.com","H"));
personas.add(new Persona("Maria de las Mercedes","mmercedes@gmail.com","M"));

//Creamos un Adaptador del tipo que creamos con Anterioridad
PersonaAdapter personaAdapter = new PersonaAdapter( this ,
R.layout.list_personas_item , personas);

//Añadimos el adaptador al ListView
lista.setAdapter(personaAdapter);
```

Resultado Final