

## Spinner dropdown

É um componente que faz a função de lista que pode ser “dropada” e escolhida pelo usuário.

Vamos criar uma lista básica e estática para colocarmos no nosso componente e o usuário selecionar o valor desejado.

Para isso vamos:

### 1. Lista os Itens na Spinner

Abra o arquivo, “**res/values/strings.xml**”, nela vamos definir os itens que vão ser apresentados no display do Spinner (dropdown list).

Arquivo: *res/values/strings.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">UllerApp</string>
    <string name="country_prompt">Selecione os professores</string>
    <string-array name="professores_arrays">
        <item>Daniel</item>
        <item>Thiagos</item>
        <item>Diego</item>
        <item>Rogério</item>
    </string-array>
</resources>
```

### 2. Spinner (DropDown List)

Abra o arquivo, “**res/layout/activity\_professor.xml**”, e adicione o componente spinner no layout.

1. No “spinner1”, o “**android:entries**” representa o conteúdo para ser apresentado no Spinner.

Arquivo : *res/layout/main.xml*

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:entries="@array/professores_arrays"
        android:prompt="@string/professores_prompt" />
    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit" />
</LinearLayout>

```

### 3. Código

Vamos ver o exemplo do código:

Arquivo : *MyAndroidAppActivity.java*

```

package ...

imports...

public class MyAndroidAppActivity extends Activity {

    private Spinner spinner1;

    private Button btnSubmit;

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

        spinner1 = (Spinner) findViewById(R.id.spinner1);
    }
}

```

```

spinner1.setOnItemSelectedListener(new CustomOnItemSelectedListener());

btnSubmit = (Button) findViewById(R.id.btnSubmit);

btnSubmit.setOnClickListener(new OnClickListener() {

    @Override

    public void onClick(View v) {

        Toast.makeText(MyAndroidAppActivity.this,

            "OnClickListener : " +

            "\nSpinner 1 : "+ String.valueOf(spinner1.getSelectedItem()),

            Toast.LENGTH_SHORT).show();

    }

});

}

}

```

*File : CustomOnItemSelectedListener.java*

```

package ....

imports ...

public class CustomOnItemSelectedListener implements OnItemSelectedListener {

    public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {

        Toast.makeText(parent.getContext(),

            "OnItemSelectedListener : " +

            parent.getItemAtPosition(pos).toString(),

            Toast.LENGTH_SHORT).show();

    }

}

```

```
@Override
```

```
public void onNothingSelected(AdapterView<?> arg0) {  
  
    // TODO Auto-generated method stub  
  
}  
  
}
```

O ArrayAdapter.

É uma classe responsável por fazer uma ligação quando nossa fonte de dados é um array de objetos.

Note que, mesmo usando um array de Strings, podemos usar um array, já que essa classe adaptadora utiliza o método toString() para consultar qual o texto que será exibido na lista.

Papar utilizarmos tem que passar no seu construtor o contexto, o layout e o array desejados;

```
private ArrayAdapter<String> mAdaptador;  
  
mAdaptador = new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,  
professores);
```

Vamos criar um array de String:

```
private static final String[] professores = new String[]  
{ "Daniel", "Thiagos", "Diego", "Rogério" };
```

Por último, precisamos “ligar” a lista com o nosso adaptador. Para isso, escrevemos o código:

```
setListAdapter(mAdaptador);
```