

CS 475

ToolBar with
Spinner
Checkbox
EditText
Data Adapter

Tool Bar- Custom Widgets

What about if you want to embed a spinner?

What is a spinner?

- A combo box, with choices
- Quick way to select one value from a set



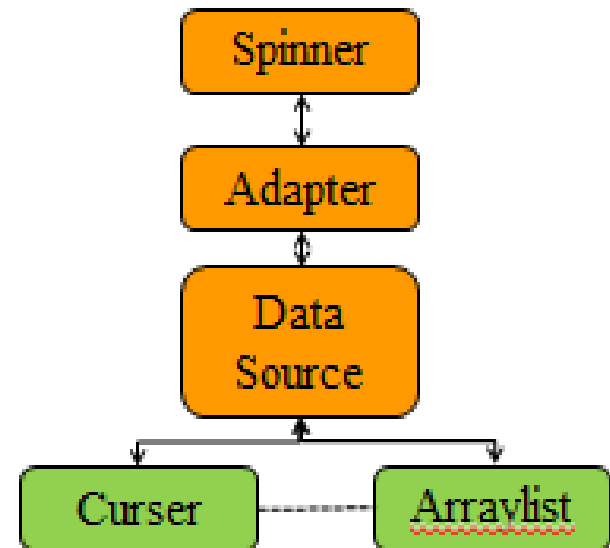
Spinners

Steps to add a spinner

- Add Spinner to ToolBar
- Create Data Adapter and layouts (*list of selectable items for the drop-down and the layout to use when drawing each item in the list*)
- Bind the Data Adapter to Spinner
- Create event handler to respond to user events

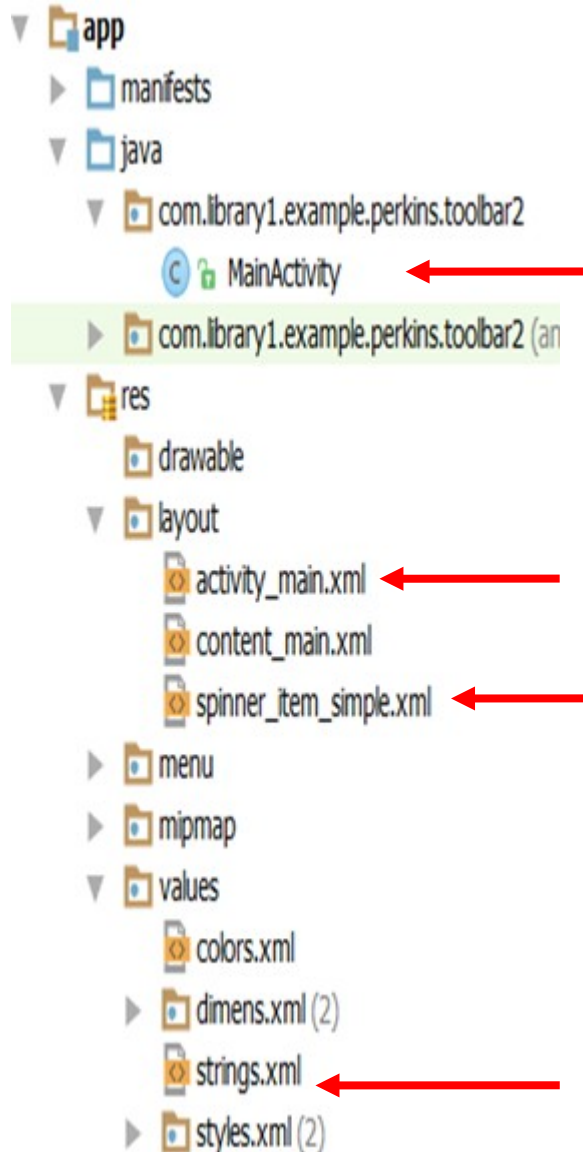
Data Adapters

- Manages datasource for a view
- Consistent access protocol
- Easy datasource swapping
- Used by a lots of things
 - ListView
 - Spinner
 - Contacts list



- **Standard** and Custom Adapter

Spinners

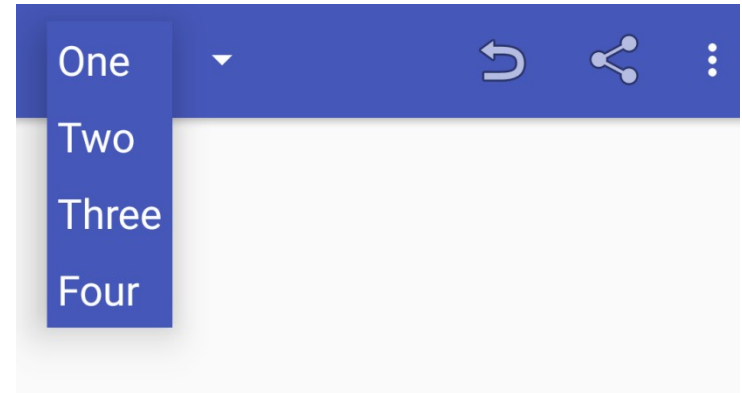


Create adapter,
Bind to spinner
set listener

Where ToolBar and
extra widgets go

Spinner row
appearance

For this example, a simple array of options



Spinners- activity_main.xml

```
<android.support.v7.widget.Toolbar  
    android:id="@+id/toolbar"  
    android:layout_width="match_parent"  
    android:layout_height="?attr/actionBarSize"  
    android:background="?attr/colorPrimary"  
    app:popupTheme="@style/AppTheme.PopupOverlay">
```

```
<Spinner  
    android:id="@+id/spinner"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingRight="10dp"/>
```

```
<CheckBox  
    android:id="@+id/checkBox"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New CheckBox"  
    android:paddingRight="10dp"  
    android:paddingLeft="10dp"/>
```

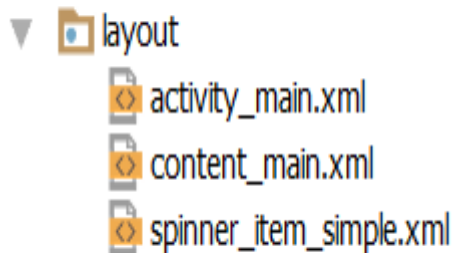
```
<EditText  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="New Text here"  
    android:id="@+id/textView"  
    android:paddingRight="10dp"/>
```

```
</android.support.v7.widget.Toolbar>
```

Where you put
extra views or
widgets, in
addition to the
menu items
found in
res/menu/main_menu.xml

Complete
Toolbar

Spinner and Row layout



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<TextView
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:textSize="20dip"
```

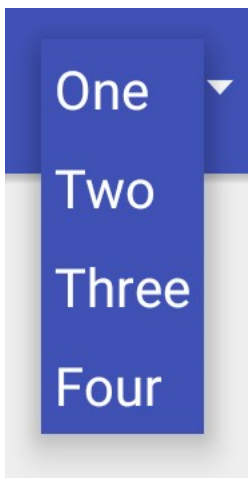
```
android:gravity="left"
```

```
android:textColor="@color/abc_primary_text_material_dark"
```

```
android:background="@color/colorPrimary"
```

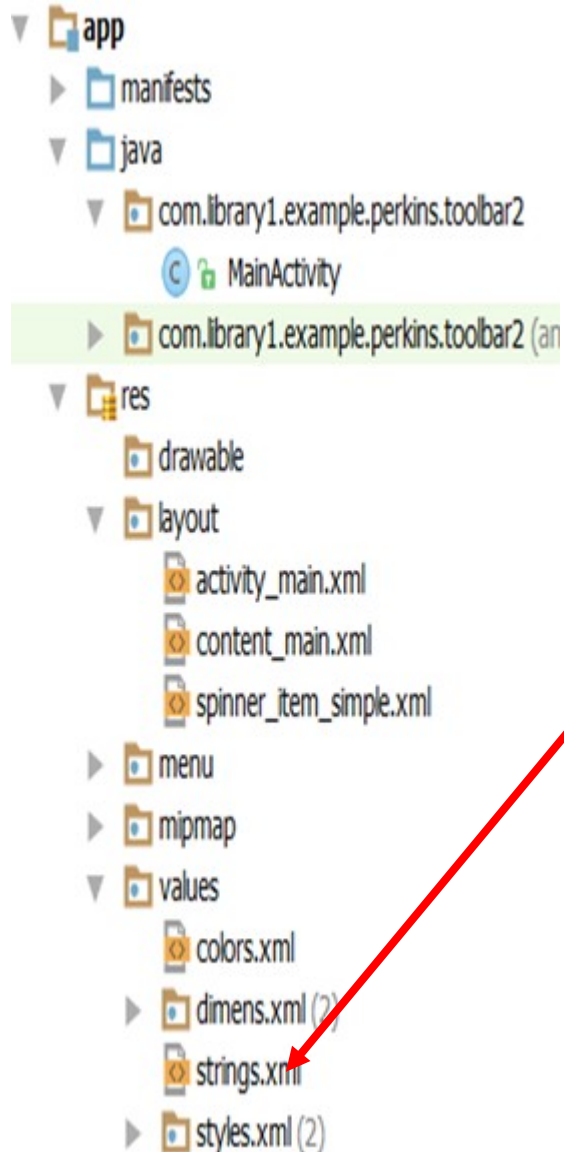
```
android:padding="5dip"
```

```
/>
```



Defines what
each
row in the
spinner
looks like

Spinner - Data for Adapter



```
<resources>
    <string name="app_name">toolbar2</string>
    <string name="action_settings">Settings</string>
    <string-array name="numbers">
        <item>One</item>
        <item>Two</item>
        <item>Three</item>
        <item>Four</item>
    </string-array>
</resources>
```


Spinner - Java - Setup

```
Spinner spinner;
```

```
private void setupSimpleSpinner() {
```

```
    //create a data adapter to fill above spinner with choices
```

```
    //R.array.numbers is arraylist in strings.xml
```

```
    //R.layout.spinner_item_simple is just a textview
```

```
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.numbers, R.layout.spinner_item_simple);
```

```
    //get a reference to the spinner
```

```
    spinner = (Spinner) findViewById(R.id.spinner);
```

```
    //bind the spinner to the datasource managed by adapter
```

```
    spinner.setAdapter(adapter);
```

```
    //respond when spinner clicked
```

```
    spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {
```

```
        public static final int SELECTED_ITEM = 0;
```

```
        @Override
```

```
        public void onItemClick(AdapterView<?> arg0, View arg1, int pos, long rowid) {
```

```
            if (arg0.getChildAt(SELECTED_ITEM) != null) {
```

```
                ((TextView) arg0.getChildAt(SELECTED_ITEM)).setTextColor(Color.WHITE);
```

```
                Toast.makeText(MainActivity.this, (String) arg0.getItemAtPosition(pos), Toast.LENGTH_SHORT).show();
```

```
            }
```

```
        }
```

```
        @Override
```

```
        public void onNothingSelected(AdapterView<?> arg0) {
```

```
        }
```

```
    });
```

```
}
```

Create simple adapter



Get reference to spinner

Bind Adapter

Handle onClick
Events in Spinner



Checkbox, EditText

```
<android.support.v7.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="?attr/actionBarSize"
    android:background="?attr/colorPrimary"
    app:popupTheme="@style/AppTheme.PopupOverlay">
```

```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingRight="10dp"/>
```

```
<CheckBox
    android:id="@+id/checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New CheckBox"
    android:paddingRight="10dp"
    android:paddingLeft="10dp"/>
```

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Text here"
    android:id="@+id/textView"
    android:paddingRight="10dp"
    android:ems="4" />
```

```
</android.support.v7.widget.Toolbar>
```

```
EditText myText;
```

```
private void setupEditText() {
    myText = (EditText)findViewById(R.id.textView);
    myText.setText("New Text");
}
```

```
CheckBox myCheckbox;
```

```
private void setupCheckBox() {
    myCheckbox = (CheckBox)findViewById(R.id.checkbox);
    myCheckbox.setOnCheckedChangeListener((buttonView, isChecked) → {
        String myString = (isChecked)? "Checked": "UN Checked";
        Toast.makeText(MainActivity.this, myString, Toast.LENGTH_SHORT).show();
    });
}
```

Limits to max of 4 widest chars

Summary

- Create a toolbar
- With icons (and text if you want)
- And widgets (spinners, edittexts, checkboxes)
 - (Really any widget if you follow recipes outlined above)
- A bit about data adapters
 - decouple datasource from app
 - Consistent API