

Spinner

[Spinner](#) is a widget similar to a drop-down list for selecting items.

In this tutorial, you'll create a simple spinner widget that displays a list of planets. When one is selected, a toast message will display the selected item.

1. Start a new project named *HelloSpinner*.
2. Open the `res/layout/main.xml` file and insert the following:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="10dip"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dip"
        android:text="@string/planet_prompt"
    />
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:prompt="@string/planet_prompt"
    />
</LinearLayout>
```

Notice that the [TextView](#)'s `android:text` attribute and the [Spinner](#)'s `android:prompt` attribute both reference the same string resource. This text behaves as a title for the widget. When applied to the [Spinner](#), the title text will appear in the selection dialog that appears upon selecting the widget.

3. Create a `strings.xml` file in `res/values/` and edit the file to look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="planet_prompt">Choose a planet</string>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
        <item>Jupiter</item>
        <item>Saturn</item>
        <item>Uranus</item>
        <item>Neptune</item>
    </string-array>
</resources>
```

The `<string>` element defines the title string referenced by the [TextView](#) and [Spinner](#) in the layout above. The `<string-array>` element defines the list of strings that will be displayed as the list in the [Spinner](#) widget.

4. Now open the `HelloSpinner.java` file and insert the following code for the `onCreate()` method:

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    Spinner spinner = (Spinner) findViewById(R.id.spinner);
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
        this, R.array.planets_array, android.R.layout.simple_spinner_item);
    adapter.setDropDownViewResource
        (android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
}
```

After the `main.xml` layout is set as the content view, the [Spinner](#) widget is captured from the layout with `findViewById(int)`. The `createFromResource()` method then creates a new [ArrayAdapter](#), which binds each item in the string array to the initial appearance for the [Spinner](#) (which is how each item will appear in the spinner when selected). The `R.array.planets_array` ID references the `string-array` defined above and the `android.R.layout.simple_spinner_item` ID references a layout for the standard spinner appearance, defined by the platform. Then `setDropDownViewResource(int)` is called to define the appearance for each item when the widget is opened (`simple_spinner_dropdown_item` is another standard layout defined by the platform). Finally, the [ArrayAdapter](#) is set to associate all of its items with the [Spinner](#) by calling `setAdapter(T)`.

5. Now create a nested class that implements [AdapterView.OnItemSelectedListener](#). This will provide a callback method that will notify your application when an item has been selected from the [Spinner](#). Here's what this class should look like:

```
public class MyOnItemSelectedListener implements OnItemSelectedListener {

    public void onItemSelected(AdapterView<?> parent,
        View view, int pos, long id) {
        Toast.makeText(parent.getContext(), "The planet is " +
            parent.getItemAtPosition(pos).toString(), Toast.LENGTH_LONG).show();
    }

    public void onNothingSelected(AdapterView parent) {
        // Do nothing.
    }
}
```

The [AdapterView.OnItemSelectedListener](#) requires the `onItemSelected()` and `onNothingSelected()` callback methods. The former is called when an item from the [AdapterView](#) is selected, in which case, a short [Toast](#) message displays the selected text; and the latter is called when a selection disappears from the [AdapterView](#), which doesn't happen in this case, so it's ignored.

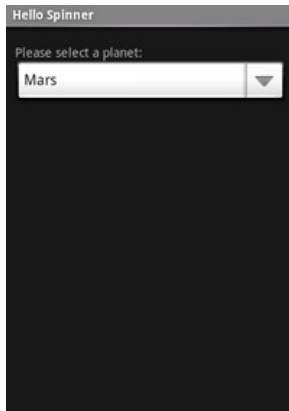
6. Now the `MyOnItemSelectedListener` needs to be applied to the [Spinner](#). Go back to the `onCreate()` method and add the following line to the end:

```
spinner.setOnItemSelectedListener(new MyOnItemSelectedListener());
```

This creates a new anonymous instance of the `MyOnItemSelectedListener` and sets it as the listener for the [Spinner](#).

7. Run the application.

It should look like this:



Resources

- [R.layout](#)
- [ArrayAdapter](#)
- [Spinner](#)

[← Back to Hello, Views](#)

[↑ Go to top](#)

Except as noted, this content is licensed under [Creative Commons Attribution 2.5](#). For details and restrictions, see the [Content License](#).

[Site Terms of Service](#) - [Privacy Policy](#) - [Brand Guidelines](#)