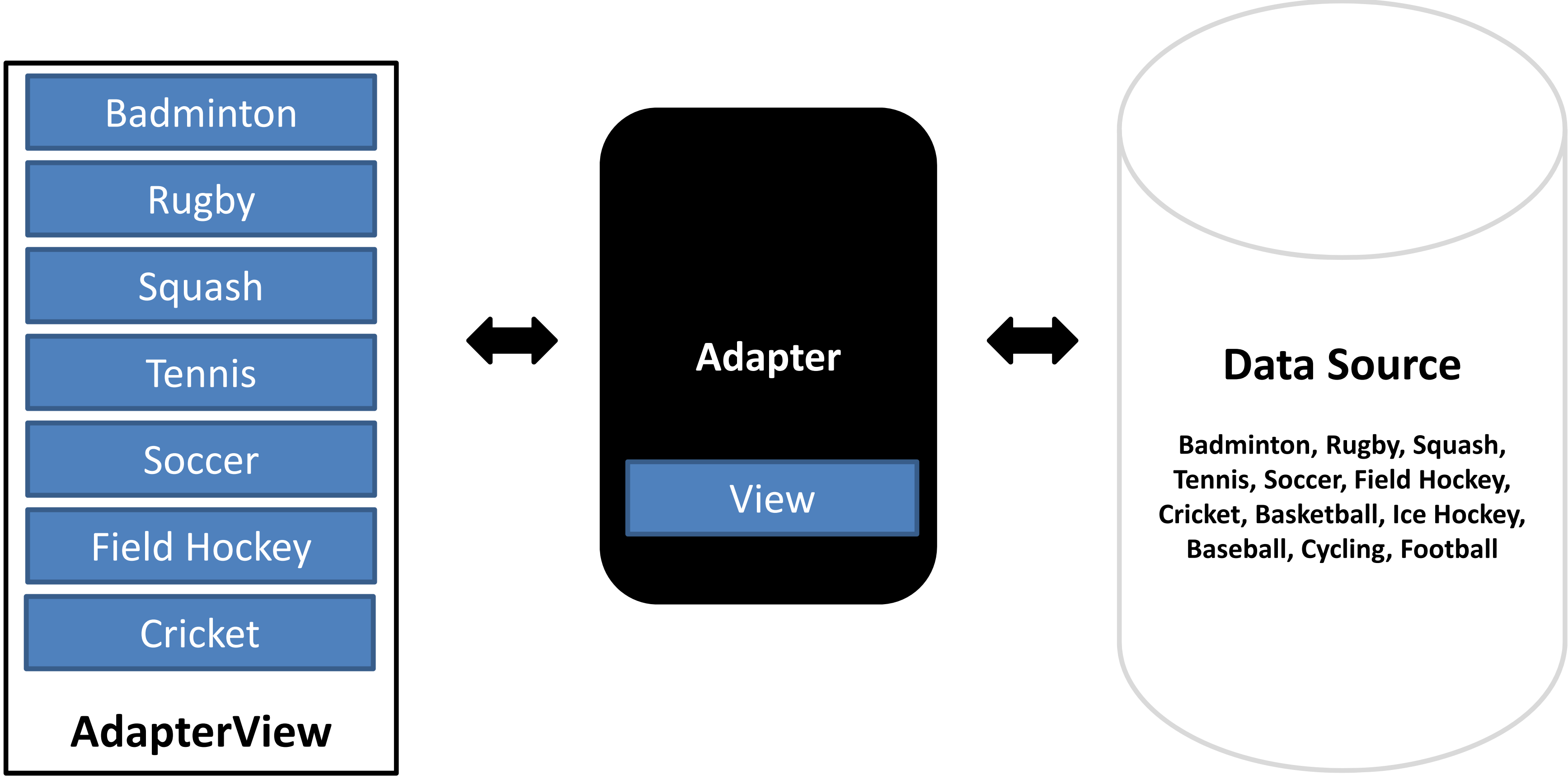
Mobile Application Development

Adapters

Adapter

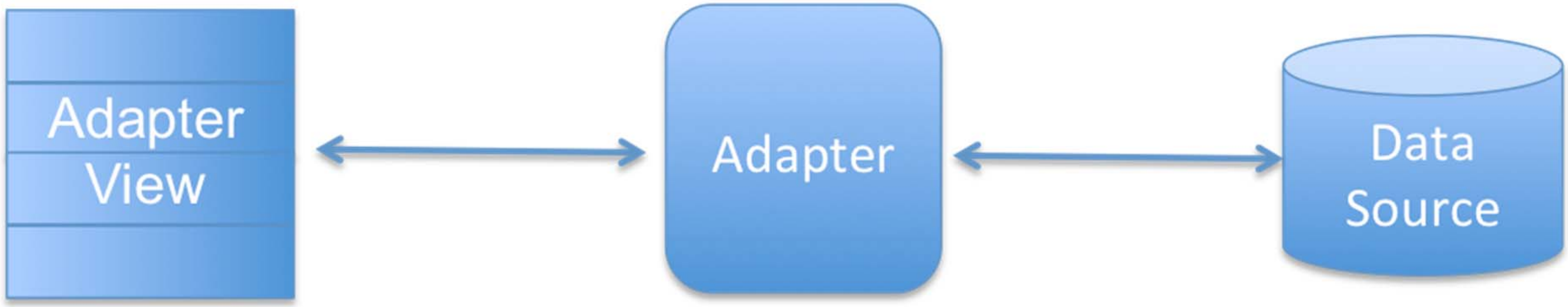


Badminton
Rugby
Squash
Tennis
Soccer
Field Hockey
Cricket
AdapterView
Adapter
View
Data Source
Badminton, Rugby, Squash,
Tennis, Soccer, Field Hockey,
Cricket, Basketball, Ice Hockey,
Baseball, Cycling, Football

Adapter

- An Adapter object acts as a bridge between an AdapterView and the underlying data for that view.
- The Adapter provides access to the data items.
- The Adapter is also responsible for making a View for each item in the data set.

Adapter



- List View
- Grid View
- Spinner

- ArrayAdapter
- SimpleCursorAdapter
- BaseAdapter

- Array
- Database

AdapterView

Note that items in AdapterView:

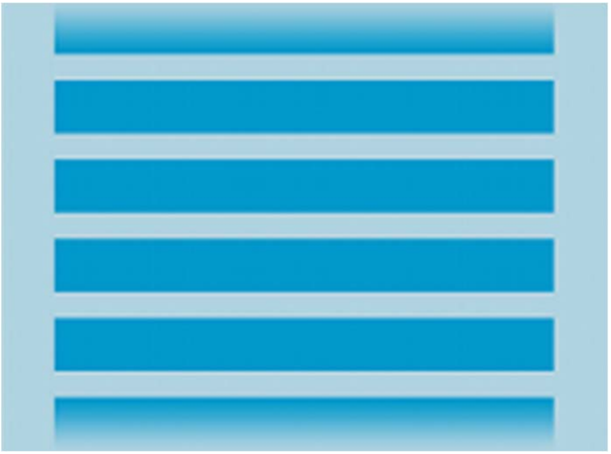
- Are of same type. i.e., TextView, ImageView, etc.
- Needs to be accessed and added at runtime from some data source e.g., array, database, etc.

Adapter & AdapterView

- **Adapter Class:**
  - Provides access to the data items.
  - Responsible for making a View for each item in the data set.
- **AdapterView Class:**
  - Whose children are determined by an Adapter.
  - Provide mechanism to handle click event on particular list items.
  - Takes care of scrolling & pagination.
  - ListView & GridView are subclasses of AdapterView.

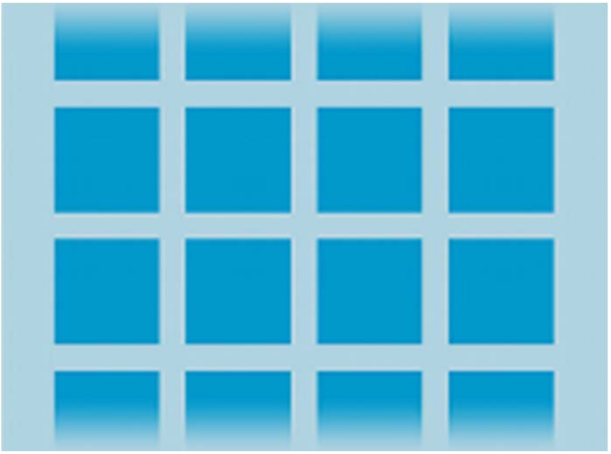
List View & Grid View

List View



Displays a scrolling single
column list.

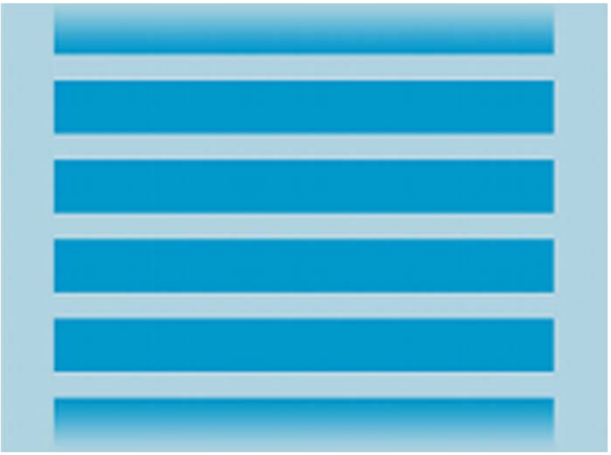
Grid View

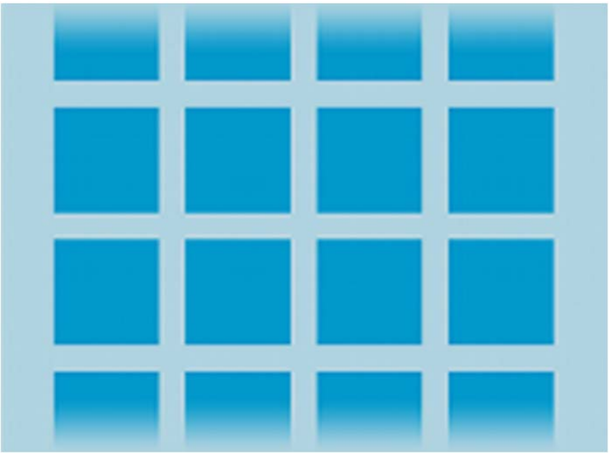


Displays a scrolling grid of
columns and rows.

List View & Grid View

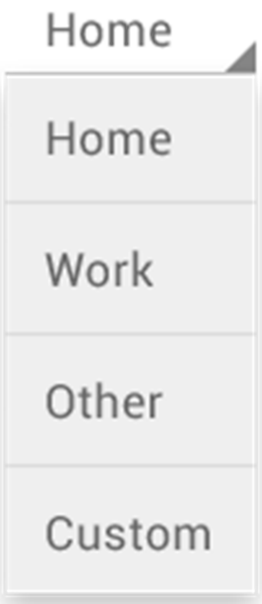
- The content for “List View” & “Grid View” is dynamic.
- These are populated with views at runtime.





Spinner

- Spinner and Gallery (Gallery has been deprecated since API 16) are also commonly used subclasses of AdapterView.



Spinners provide a quick way to select one value from a set. In the default state, a spinner shows its currently selected value. Touching the spinner displays a dropdown menu with all other available values, from which the user can select a new one.

Using Adapter & AdapterView

1. Add AdapterView to Activity's Layout (i,e., Add in XML Layout)

2. Define Date Source (e.g., Array in Code or XML "string-array")

3. Create Adapter (e.g., ArrayAdapter)

   – Specify context, layout to be used for each view, and the string array

4. Get reference to AdapterView (use findViewById() method)

5. Set Adapter on Adapter View (use setAdapter() method)

6. Set Event Listener (if needed. e.g., onItemClick())

ArrayAdapter Examples

- List View
- Grid View
- Spinner

EXAMPLE:
List View With Array Adapter

1. Add AdapterView

```
<RelativeLayout
   ... >

   <ListView
        android:id="@+id/listView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true" >
   </ListView>

</RelativeLayout>
```

2. Define Date Source

```
String[] months = {"January", "February", "March", "April",
   "May", "June", "July",  "August", "September", "October",
   "November", "December"};
```

3. Create Adapter

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

4. Get reference to AdapterView

```
ListView listview = (ListView) findViewById(R.id.listView1);
```

5. Set Adapter on Adapter View

```
listview.setAdapter(adapter);
```

6. Set Event Listener

```
listview.setOnItemClickListener(new OnItemClickListener()
{
   public void onItemClick(AdapterView<?> parent, View v, int
   position, long id) {

   Intent intent=new Intent(getApplicationContext(),Second.class);
   intent.putExtra("v1", String.valueOf(position));
   startActivity(intent);

   }
});
```

EXAMPLE:
Grid View With Array Adapter

1. Add AdapterView

```
<RelativeLayout
   ... >

   <GridView
        android:id="@+id/gridView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:numColumns="3" >
   </GridView>

</RelativeLayout>
```

2. Define Date Source

In “strings.xml” File:

```
<string-array name="alphabets_array">
   <item>A</item>
   <item>B</item>
   <item>C</item>
   . . .
   <item>Z</item>
</string-array>
```

2. Define Date Source

In Activity’s “.java” File:

```
// You can declare outside method, But don’t initialize
String[] alphabets;



// Get values from XML resource file
alphabets =
   getResources().getStringArray(R.array.alphabets_array);
```

3. Create Adapter

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

4. Get reference to AdapterView

```
GridView gridview = (GridView) findViewById(R.id.gridView1);
```

5. Set Adapter on Adapter View

```
gridview.setAdapter(adapter);
```

6. Set Event Listener

```
gridview.setOnItemClickListener(new OnItemClickListener()
{
   public void onItemClick(AdapterView<?> parent, View v, int
   position, long id)
   {

   Intent intent=new Intent(getApplicationContext(),Second.class);
   intent.putExtra("v1", String.valueOf(position));
   startActivity(intent);


   }
});
```

EXAMPLE:
Spinner With Array Adapter

1. Add AdapterView

```
<RelativeLayout
   ... >

   <Spinner
       android:id="@+id/spinner1"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_alignParentLeft="true"
       android:layout_alignParentTop="true" />

   . . .
</RelativeLayout>
```

2. Define Date Source

In “strings.xml” File:

```
 <string-array name="sports_array">
        <item>Badminton</item>
        <item>Rugby</item>
        <item>Squash</item>
        <item>Tennis</item>
        <item>Soccer</item>
        <item>Field Hockey</item>
        <item>Cricket</item>
        <item>Basketball</item>
        <item>Ice Hockey</item>
        <item>Baseball</item>
        <item>Cycling</item>
        <item>Football</item>
</string-array>
```

2. Define Date Source

In Activity’s “.java” File:

```
// You can declare outside method, But don’t initialize
String[] sports;
```

```
// Get values from XML resource file
sports = getResources().getStringArray(R.array. sports_array);
```

3. Create Adapter

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
   android.R.layout.simple_spinner_dropdown_item, sports);
```

4. Get reference to AdapterView

```
Spinner spinner = (Spinner) findViewById(R.id.spinner1);
```

5. Set Adapter on Adapter View

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

6. Set Event Listener

```
spinner.setOnItemSelectedListener(new
   AdapterView.OnItemSelectedListener()
{
   public void onItemSelected(AdapterView<?> parent, View view, int
   pos, long id)
   {

   Intent intent=new Intent(getApplicationContext(),Second.class);
   intent.putExtra("v1", String.valueOf(pos));
   startActivity(intent);

   }

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

   }
});
```

How to Access Spinner Value?

```
Spinner spinner = (Spinner) findViewById(R.id.spinner1);

int selected_sport=spinner.getSelectedItemPosition();
```

References

- http://developer.android.com/reference/android/widget/ArrayAdapter.html
- http://developer.android.com/guide/topics/ui/layout/listview.html
- http://developer.android.com/guide/topics/ui/layout/gridview.html
- http://developer.android.com/guide/topics/ui/declaring-layout.html#AdapterViews
- http://developer.android.com/reference/android/widget/AutoCompleteTextView.html

Summary

- **Data Sources:** Array, Database
- **Adapters:** ArrayAdapter, SimpleCursorAdapter, BaseAdapter
- **AdapterViews:** List View, Grid View, Spinner
- **Implementation:**
  - Create Data Source
  - Add AdapterView to Activity’s Layout
  - Create Adapter
  - Get reference to AdapterView in Activity
  - Set Adapter on AdapterView
  - Set EventListener

Q & A