INTRODUCTION TO

LISTVIEWS

and adapters

TECHNOGUFF

fb.com/technoguff @technoguff

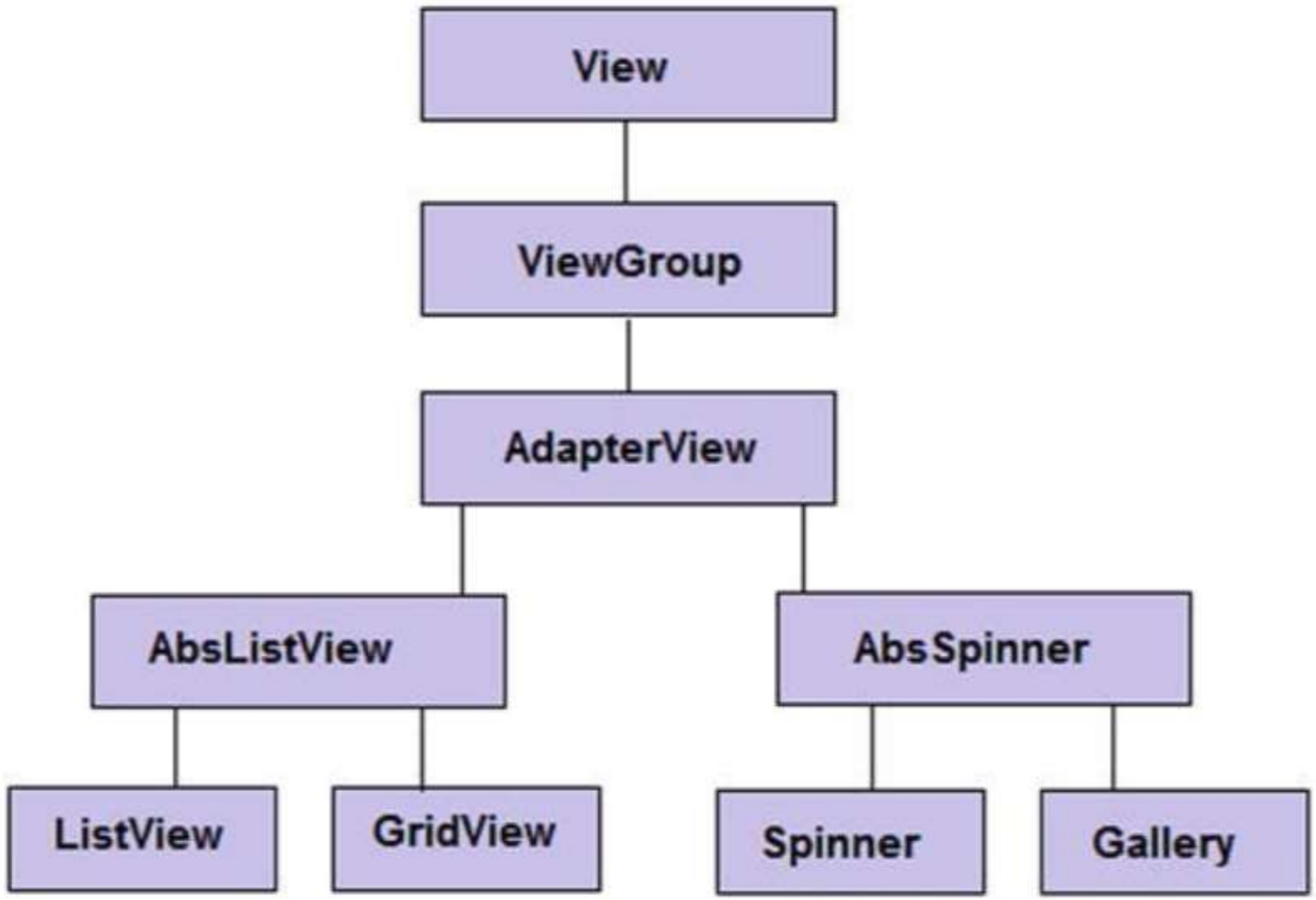
In this session

- Introduction to ListView
- Introduction to simple List Adapters

Adapter Views

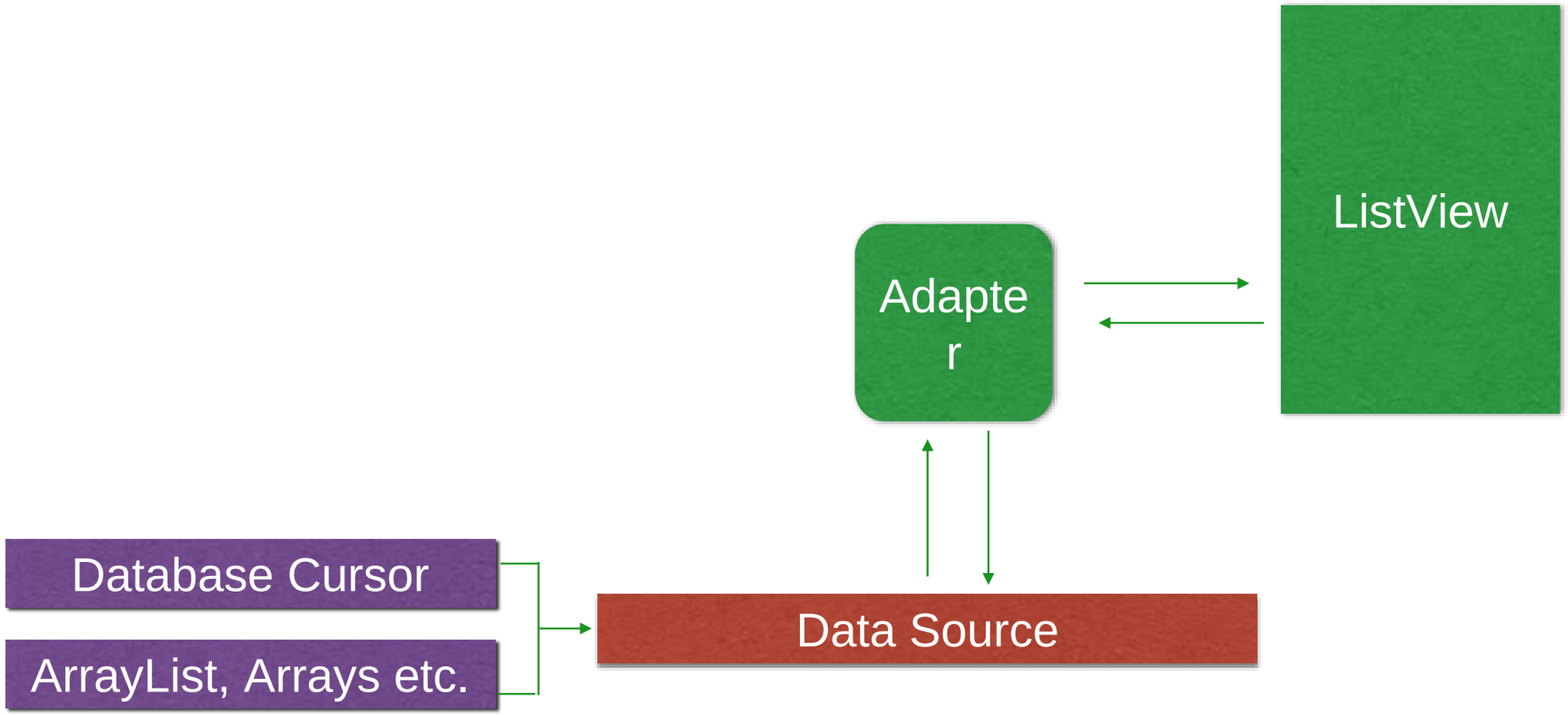
- Android widgets used to display collection of data.
- Adapter Views use adapters for managing data
- Examples of adapter views are
  - **Spinner,**
  - **ListView,**
  - **Gallery(deprecated in API level 16),**
  - **GridView.**

AdapterView Hierarchy



gallery is deprecated since API Level 16

Flow Diagram



Adapter

- An Adapter object acts as a bridge between an AdapterView and the underlying data for that view.

Adapters

- BaseAdapter
  - provides data model for list
  - converts data into fields of the list
  - extended by all adapters

- **ArrayAdapter**
- **SimpleCursorAdapter**

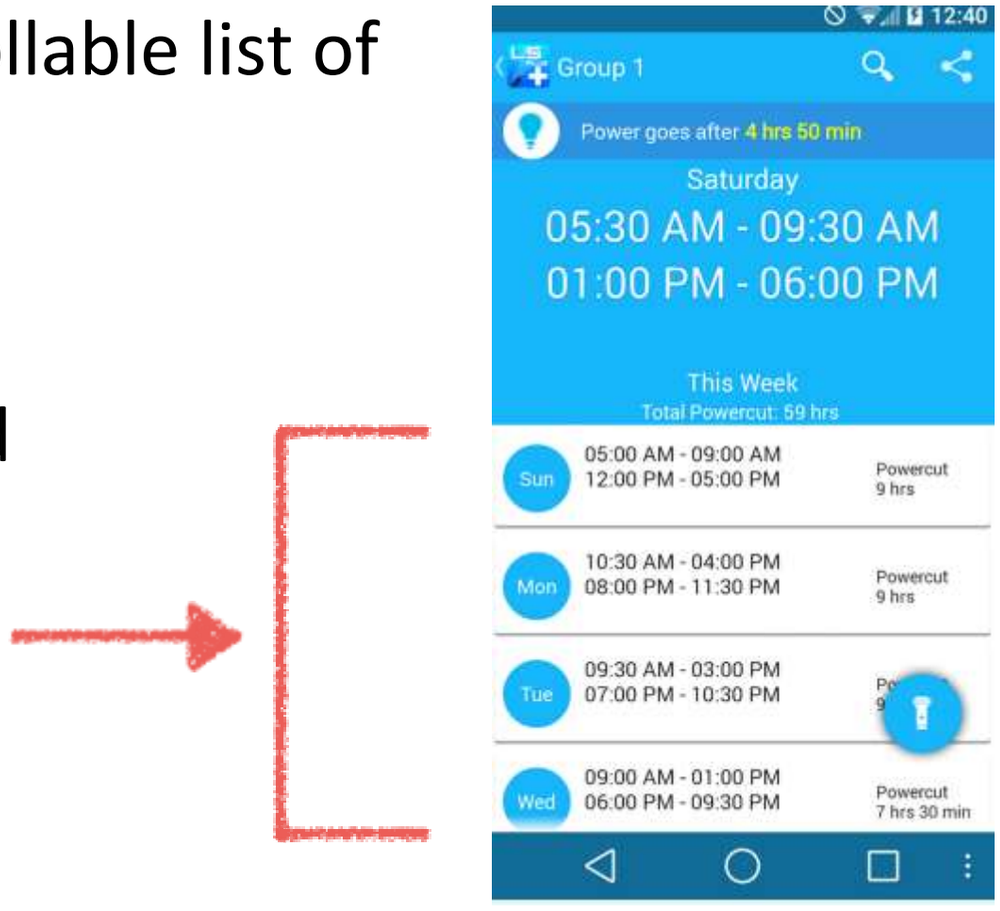
- Custom Adapter

ArrayAdapter

- handles data based on Arrays or java.util.List

ListView

- A view capable of displaying scrollable list of items
- Creates Views only when needed
- Recycles Views



ListView example from Laodshedding+ app

Choice Mode

- CHOICE_MODE_NONE
- CHOICE_MODE_SINGLE
- CHOICE_MODE_MULTIPLE

Using ListView

- Declare our ListView in our layout.xml
- Create our Adapter class
- Fetch the items for our list
- Specify the layout that we want for our list items
- Plug our Adapter with our declared Listview

Implementing ListView using ArrayAdapter

- 1) Add ListView to Activity Layout

```
<ListView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/listview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

Implementing ListView…

- prepare adapter

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

- Set adapter

```
mListView.setAdapter(adapter);
```

Using Custom Adapter

- Create a layout of custom View for list item
- Create an instance of an adapter
- Override various methods inside the adapter class such as , getView(), getCount() etc.
- Set the custom adapter to the listView.

ViewHolder Pattern

- in a long list findViewById() might be called frequently during the scrolling of ListView, which slows down the performance

- **ViewHolder** comes handy in such case.

- A **ViewHolder** object stores each of the component views inside the tag field of the Layout, so you can immediately access them without the need to look them up repeatedly.

Listeners

- setOnItemClickListener
- setOnItemLongClickListener
- setOnItemSelectedListener
- setOnScrollListener

Headers & Footers

- **addHeaderView**
  - Add a fixed view to appear at the top of the list.
- **addFooterView**
  - Add a fixed view to appear at the bottom of the list.

Adding HeaderView or FooterView

- Prepare layout for header or footer
- Inflate the layout
- add the view using addHeaderView() method

```
View header = getLayoutInflater().inflate(R.layout.header_view, mListView, false);

mListView.addHeaderView(header, null, false);
```

ListActivity

- If displaying list is primary purpose, ListActivity is used.
- displays a list of items by binding to a data source(array, cursor).
- Exposes event handlers when the user selects an item

ListActivitiy

- Simplifies the handling of ListView
- Set List Adapter in the onCreate() method using setListAdapter()

```
setListAdapter(adapter);
```

- Register click by onListItemClick()

```
@Override
protected void onListItemClick(ListView list, View view, int position, long id) {
```

Thank You

Find source code of ListView Example on

https://github.com/technoguff/ListViewExample

And tutorial at

http://blog.technoguff.com/2015/07/introduction-to-listview.html

TECHNOGUFF

fb.com/technoguff @technoguff