



# Lập trình Android

## Bài 4: *List-based widgets*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-cSDL>

2019





# Nội dung

---

1. List-based widgets
2. ScrollView
3. ListView
4. GridView
5. RecyclerView
6. Spinner

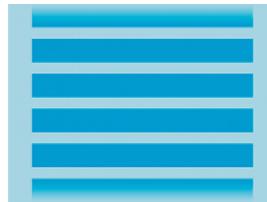




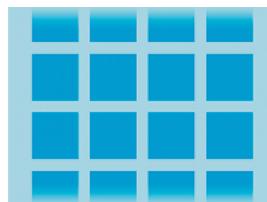
# List-based widgets

- Khi có một tập dữ liệu lớn cần trình bày, các list-based widgets thường được sử dụng

- ListView



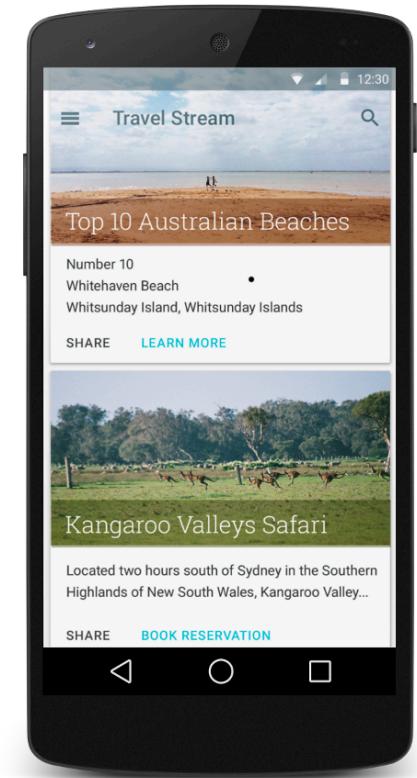
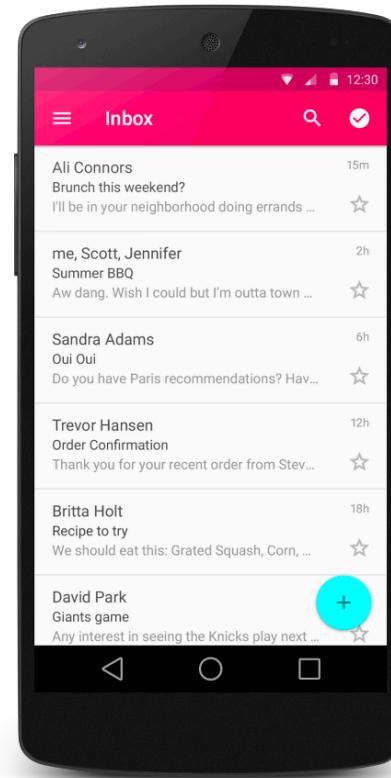
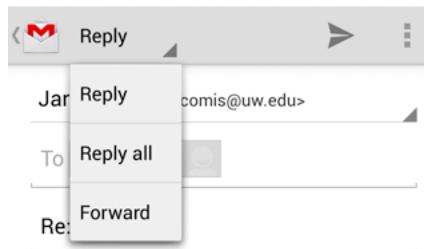
- GridView



- RecyclerView

- Spinner

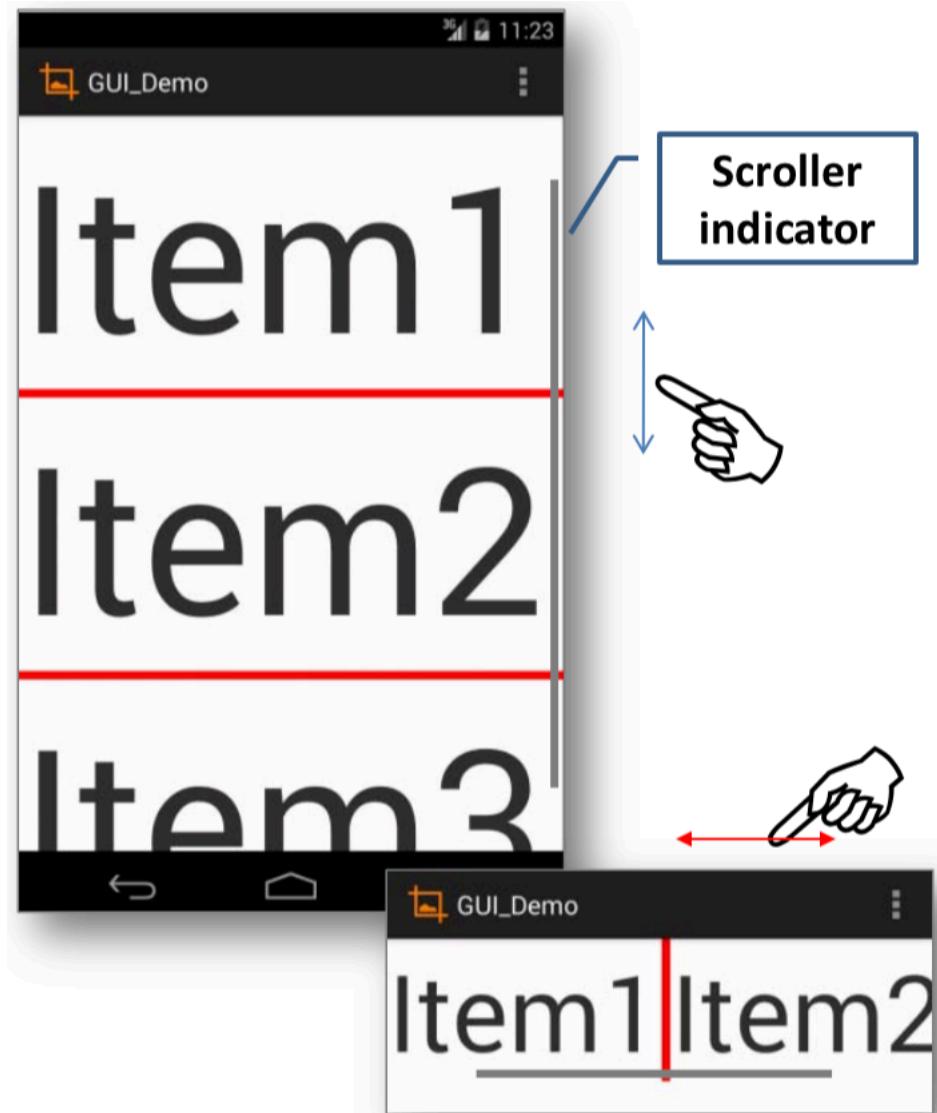
- ScrollView





# ScrollView

- ScrollView được sử dụng khi có nhiều dữ liệu cần trình bày, dài hơn so với kích thước màn hình
- **ScrollView** cung cấp khả năng cuộn theo chiều dọc
- **HorizontalScrollView** cung cấp khả năng cuộn theo chiều ngang
- Chỉ có thể thấy được 1 phần của ScrollView tại 1 thời điểm
- Bên trong 1 ScrollView bắt buộc phải có 1 LinearLayout để chứa các thành phần con và định hướng theo chiều dọc/ngang

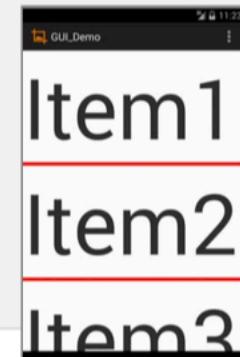




# ScrollView - Demo

```
<ScrollView  
    xmlns:android=  
        "http://schemas.android.com/apk/res/android"  
    android:id="@+id/myVerticalScrollView1"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent" >  
  
<LinearLayout  
    android:id="@+id/myLinearLayoutVertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >  
  
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Item1"  
    android:textSize="150sp" />  
  
<View  
    android:layout_width="match_parent"  
    android:layout_height="6dp"  
    android:background="#ffff0000" />
```

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Item2"  
    android:textSize="150sp" />  
  
<View  
    android:layout_width="match_parent"  
    android:layout_height="6dp"  
    android:background="#ffff0000" />  
  
<TextView  
    android:id="@+id/textView3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Item3"  
    android:textSize="150sp" />  
  
</LinearLayout>  
  
</ScrollView>
```





# HorizontalScrollView - Demo

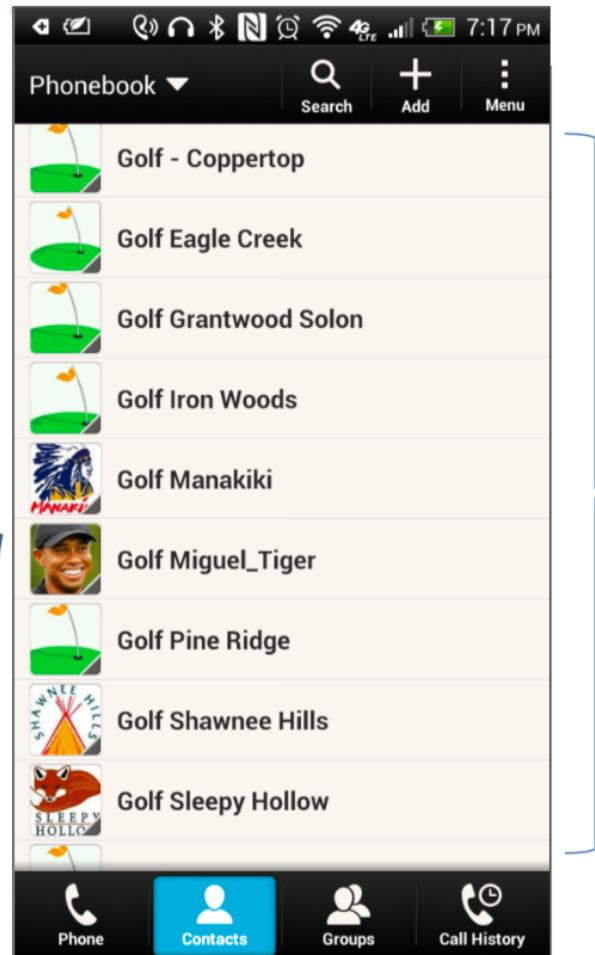
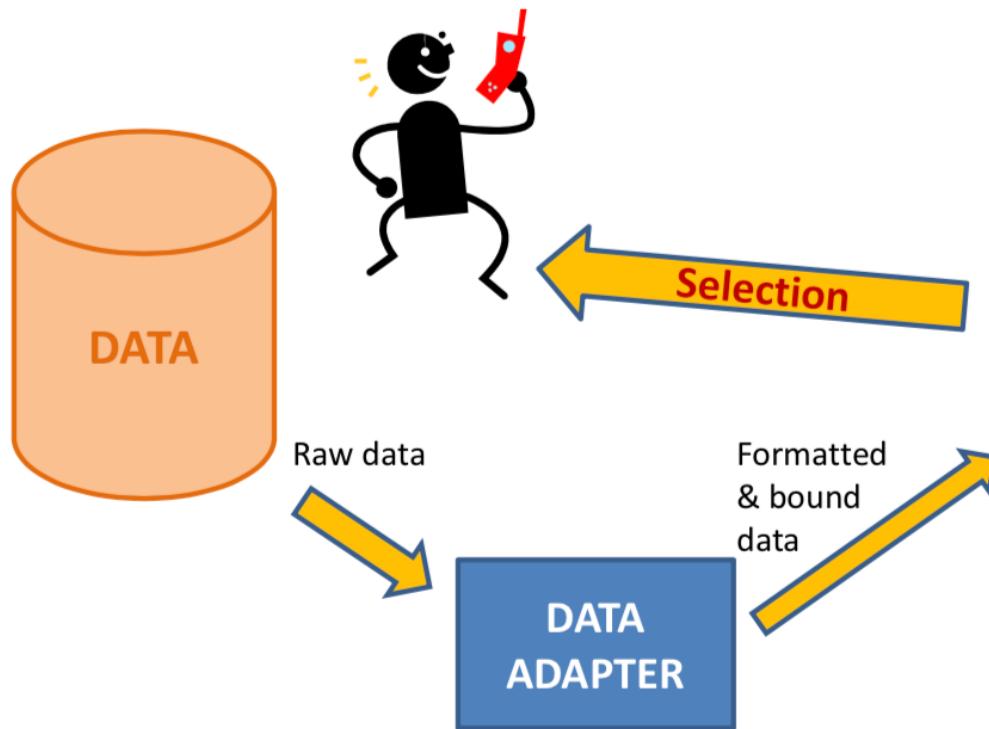
```
<HorizontalScrollView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:id="@+id/myHorizontalScrollView1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" >  
  
    <LinearLayout  
        android:id="@+id/myLinearLayoutVertical"  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:orientation="horizontal" >  
  
        <TextView  
            android:id="@+id/textView1"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="Item1"  
            android:textSize="75sp" />  
  
        <View  
            android:layout_width="6dp"  
            android:layout_height="match_parent"  
            android:background="#ffff0000" />
```

```
        <TextView  
            android:id="@+id/textView2"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="Item2"  
            android:textSize="75sp" />  
  
        <View  
            android:layout_width="6dp"  
            android:layout_height="match_parent"  
            android:background="#ffff0000" />  
  
        <TextView  
            android:id="@+id/textView3"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:text="Item3"  
            android:textSize="75sp" />  
    </LinearLayout>  
</HorizontalScrollView>
```





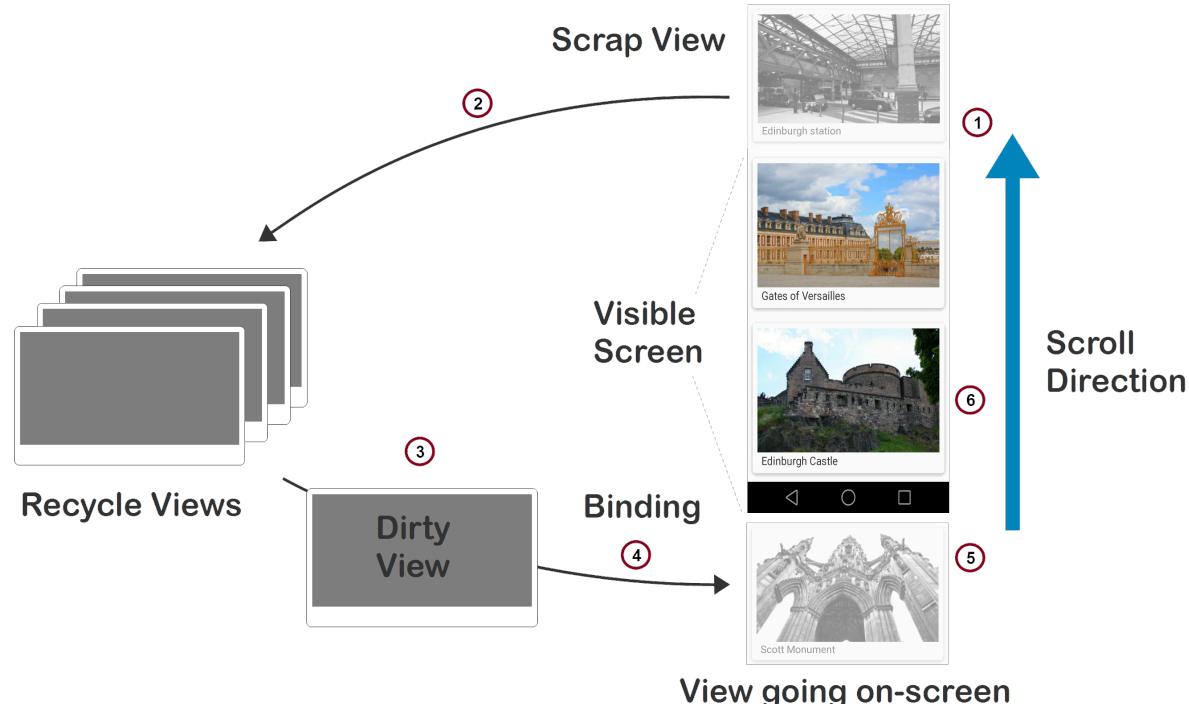
# List-based widgets



- **Data Adapter** đóng vai trò định dạng và trình bày dữ liệu
- **Data** có thể đến từ nhiều nguồn khác nhau: file, database, network...



# Tái sử dụng row view

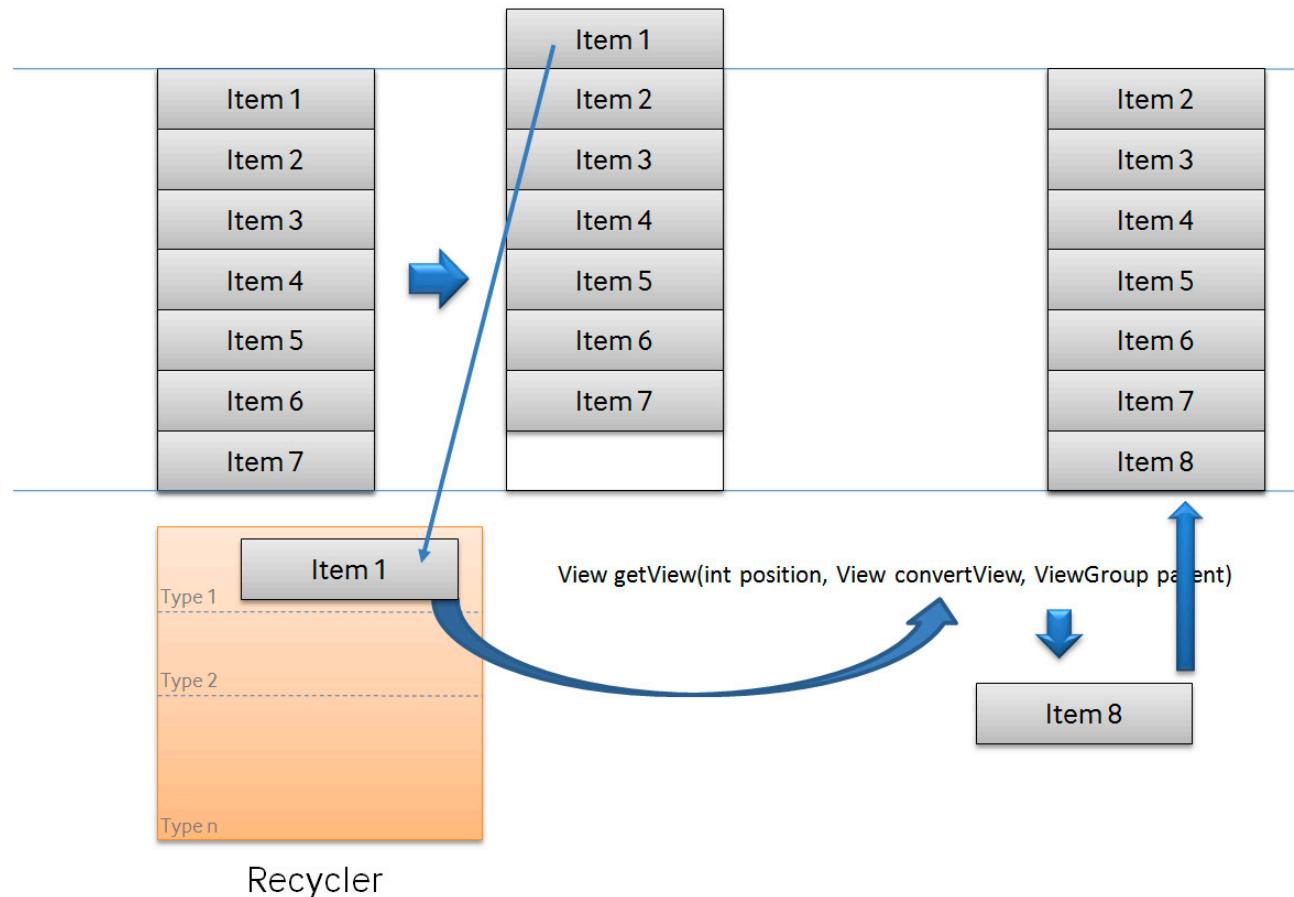


- Khi ListView kết nối đến Adapter, adapter sẽ tạo các row cho listview cho đến khi điền đầy chiều cao của listview (chiều cao của listview bị giới hạn bởi kích thước màn hình). Khi đó, sẽ không có row nào mới được tạo ra thêm trong bộ nhớ.
- Khi người dùng cuộn danh sách này, các row rời khỏi màn hình sẽ được giữ lại trong bộ nhớ để sử dụng về sau, còn các row mới đi vào màn hình sẽ tái sử dụng các row cũ còn lưu trong bộ nhớ.



## Tái sử dụng row view (2)

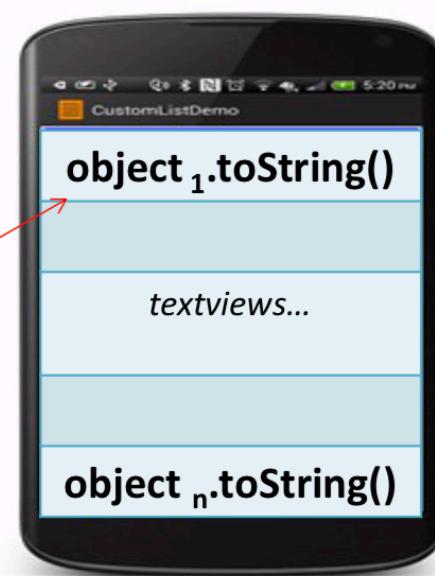
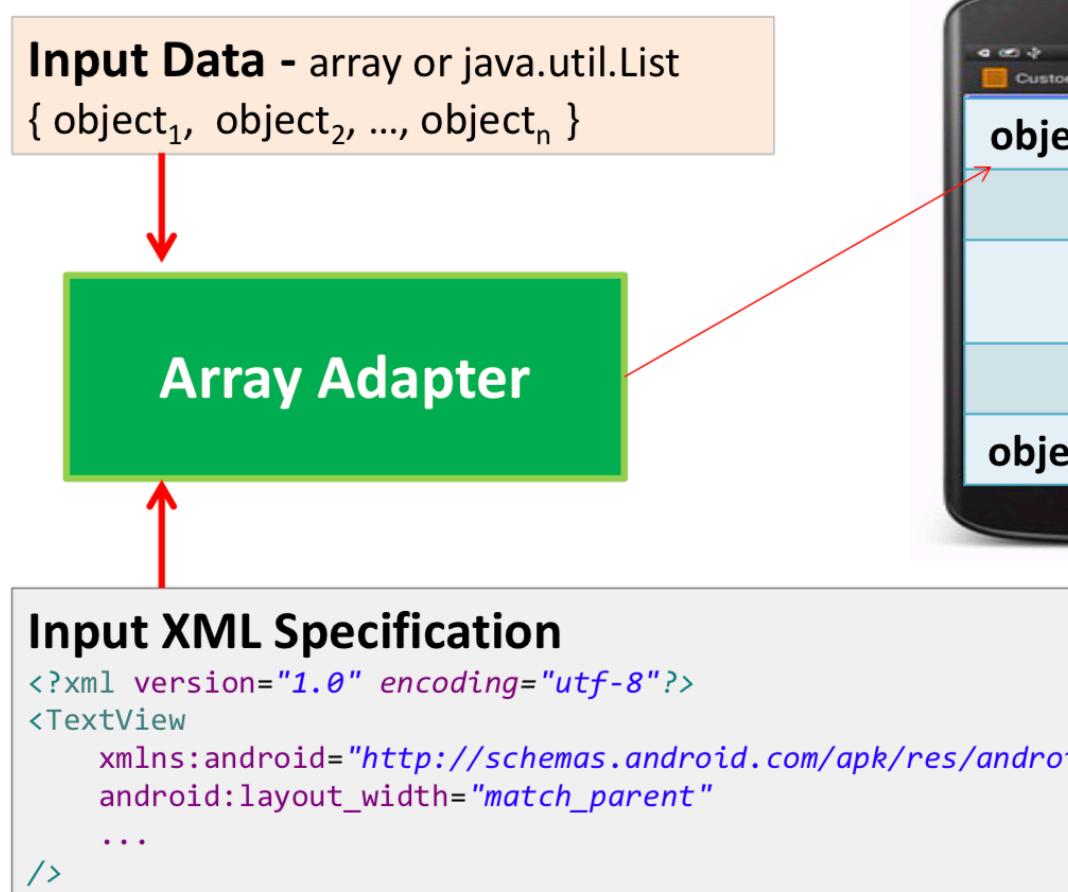
→ Dù danh sách có 1000 phần tử, chỉ khoảng 7 rows được tạo ra trong bộ nhớ (như ví dụ ở hình dưới)





# ListView

- Sử dụng ArrayAdapter để định dạng và trình bày dữ liệu cho ListView





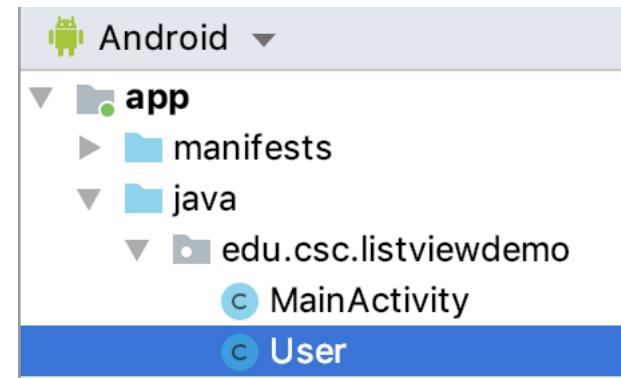
# ListView: Demo ListView với ArrayAdapter

- **Bước 1:** định nghĩa **Model**, thêm 1 class mới User như sau

```
package edu.csc.listviewdemo;

public class User {
    public String name;
    public String hometown;

    public User(String name, String hometown) {
        this.name = name;
        this.hometown = hometown;
    }
}
```



- Đây chính là đối tượng được hiển thị tương ứng với từng dòng (row) của ListView

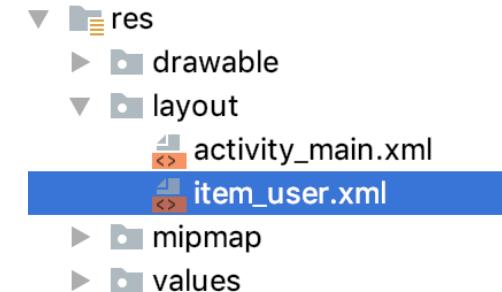
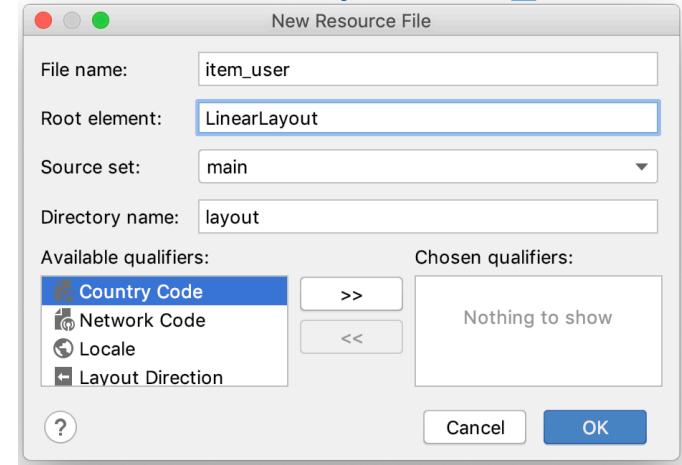


# ListView

- Bước 2: định nghĩa XML layout template cho 1 row của ListView

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@android:color/white"
    android:padding="@dimen/largeSpace">
    <TextView
        android:id="@+id/tvName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="@dimen/largeFontSize"
        android:textColor="@color/colorPrimaryDark"
        android:textStyle="bold"
        android:text="Name" />
    <TextView
        android:id="@+id/tvHome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="@dimen/mediumFontSize"
        android:textColor="@color/colorPrimary"
        android:layout_marginTop="@dimen/mediumSpace" ←→
        android:text="HomeTown" />
</LinearLayout>
```

→ thêm 1 file res/layout/item\_user.xml





# ListView

- **Bước 3: định nghĩa Adapter**

- File UserAdapter.java

```
public class UsersAdapter extends ArrayAdapter<User> {  
    public UsersAdapter(Context context, ArrayList<User> users) {  
        super(context, 0, users);  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
        // Lấy đối tượng dữ liệu tương ứng với vị trí dòng  
        User user = getItem(position);  
        // Kiểm trong view của row hiện tại có đang được tái sử dụng hay không, nếu không phải tạo mới  
        if (convertView == null)  
            convertView = LayoutInflater.from(getContext()).inflate(R.layout.item_user, parent, false);  
        // Kết nối các đối tượng XML element với Java code  
        TextView tvName = convertView.findViewById(R.id.tvName);  
        TextView tvHome = convertView.findViewById(R.id.tvHome);  
        // Gắn dữ liệu lên row view  
        tvName.setText(user.name);  
        tvHome.setText(user.hometown);  
        // Trả về row view đã xử lý hoàn tất để render lên màn hình  
        return convertView;  
    }  
}
```

Hàm **getView** giúp convert 1 đối tượng dữ liệu (Java object) thành 1 row view của ListView. Tham số **position** cho biết index của đối tượng tương ứng trong danh sách dữ liệu.



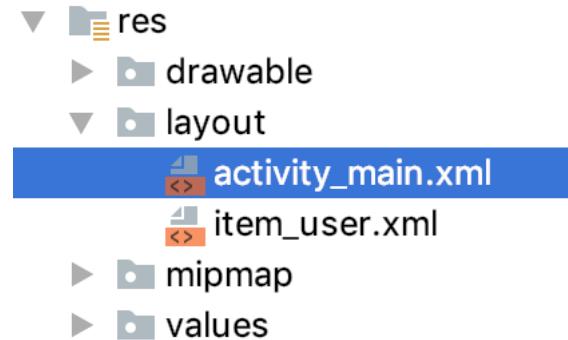
# ListView

- **Bước 4:** Khai báo ListView (activity\_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/lvItems"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```



`<ListView
 android:id="@+id/lvItems"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 android:divider="@color/colorGray"
 android:dividerHeight="@dimen/tinySpace"/>`

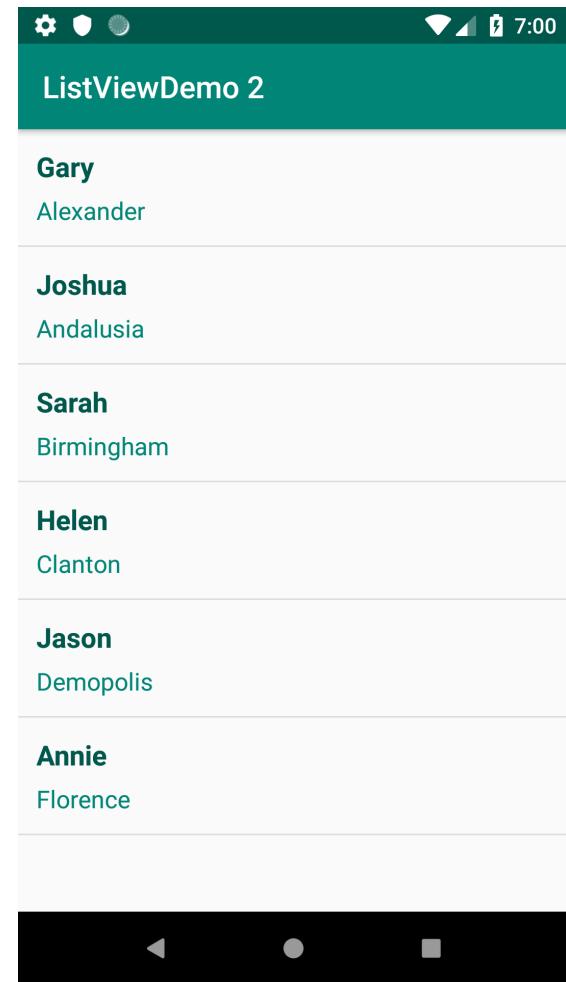
Có thể dùng thuộc tính **divider** và **dividerHeight** để thay đổi phân cách mặc định giữa các row của ListView



# ListView

- **Bước 5: Gắn Adapter vào ListView**

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        // Xây dựng nguồn dữ liệu (danh sách)  
        ArrayList<User> arrayOfUsers = getMockupData();  
        // Tạo đối tượng adapter tương ứng với danh sách dữ liệu  
        UsersAdapter adapter = new UsersAdapter(this, arrayOfUsers);  
        // Gắn adapter vào ListView  
        ListView listView = (ListView) findViewById(R.id.lvItems);  
        listView.setAdapter(adapter);  
    }  
  
    public ArrayList<User> getMockupData() {  
        // ...  
    }  
}
```





# ListView

---

- Dữ liệu có thể đến từ nhiều nguồn khác nhau (memory, file, database, network...), trong minh họa trên có thể tạo ra dữ liệu mẫu như sau:

```
public ArrayList<User> getMockupData() {  
    String[] names = {"Gary", "Joshua", "Sarah", "Helen", "Jason", "Annie"};  
    String[] hometowns = {"Alexander City", "Andalusia", "Birmingham", "Clanton", "Demopolis", "Florence"};  
  
    ArrayList<User> users = new ArrayList<>();  
    for (int i = 0; i < names.length; i++) {  
        User user = new User(names[i], hometowns[i]);  
        users.add(user);  
    }  
    return users;  
}
```



# ListView: Sử dụng ViewHolder để tăng hiệu năng xử lý

- Để tăng hiệu năng xử lý (mượt hơn), adapter cần áp dụng **ViewHolder** để **cache** việc tìm kiếm row view, giúp nạp row view nhanh hơn

```
public class UsersAdapter extends ArrayAdapter<User> {  
    // View lookup cache – Bổ sung thêm class ViewHolder  
    private static class ViewHolder {  
        TextView name;  
        TextView home;  
    }  
  
    public UsersAdapter(Context context, ArrayList<User> users) {  
        super(context, R.layout.item_user, users);  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
        // ...  
    }  
}
```



## ListView: Sử dụng ViewHolder để tăng hiệu năng xử lý (2)

- Phương thức `getView(...)`

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    User user = getItem(position);
    ViewHolder viewHolder;
    if (convertView == null) { // Kiểm tra có row view nào đang được tái sử dụng không?
        // Nếu không có row view nào được tái sử dụng, tạo 1 row view hoàn toàn mới
        viewHolder = new ViewHolder();
        LayoutInflator inflater = LayoutInflator.from(getContext());
        convertView = inflater.inflate(R.layout.item_user, parent, false);
        viewHolder.name = (TextView) convertView.findViewById(R.id.tvName);
        viewHolder.home = (TextView) convertView.findViewById(R.id.tvHome);
        // Cache đối tượng viewHolder vào trong chính row view mới này, nhờ vào tag
        convertView.setTag(viewHolder);
    } else {
        // Nếu row view được tái sử dụng, lấy đối tượng viewHolder từ tag
        viewHolder = (ViewHolder) convertView.getTag();
    }
    // Gán dữ liệu cho row view thông qua đối tượng viewHolder
    viewHolder.name.setText(user.name);
    viewHolder.home.setText(user.hometown);
    return convertView;
}
```

Việc `findViewById()` thường mất thời gian (chậm) → Nhiệm vụ của ViewHolder là **cache** việc `findViewById()`



# ListView: Xử lý sự kiện chọn 1 row của ListView

- **Xử lý sự kiện chọn 1 cell của GridView**

- Cho MainActivity *implements* interface **AdapterView.OnItemClickListener** giúp MainActivity có khả năng lắng nghe sự kiện người dùng chọn 1 cell trên GridView

```
public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener
```

- → Bắt buộc *override* phương thức **onItemClick(AdapterView<?> parent, View view, int position, long id)** tương ứng. Phương thức này được gọi khi người dùng chọn 1 cell trên GridView
- Tham số **position** của hàm **onItemClick** cho biết index của cell được người dùng chọn

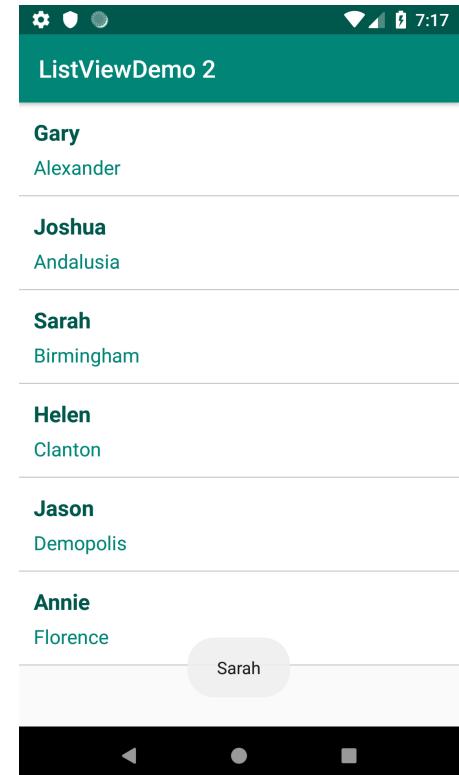
```
@Override  
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
    //...  
}
```



## ListView: Xử lý sự kiện chọn 1 row của ListView (2)

- Gán đối tượng **onItemClickListener** cho ListView chính là MainActivity

```
public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {  
    ArrayList<User> arrayOfUsers;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        arrayOfUsers = getMockupData();  
        UsersAdapter adapter = new UsersAdapter(this, arrayOfUsers);  
        ListView listView = (ListView) findViewById(R.id.lvItems);  
        listView.setAdapter(adapter);  
        → listView.setOnItemClickListener(this);  
    }  
  
    public ArrayList<User> getMockupData() {  
        //...  
    }  
  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        Toast.makeText(this, arrayOfUsers.get(position).name, Toast.LENGTH_SHORT).show();  
    }  
}
```





# ListView – Một số thao tác trên dữ liệu

---

- Thêm 1 phần tử mới vào ListView

```
User newUser = new User("Nathan", "San Diego");  
adapter.add(newUser);
```

- Thêm 1 danh sách vào ListView

```
ArrayList<User> newUsers = User.getNewMockData();  
adapter.addAll(newUsers);
```

- Xoá một phần tử xong ListView

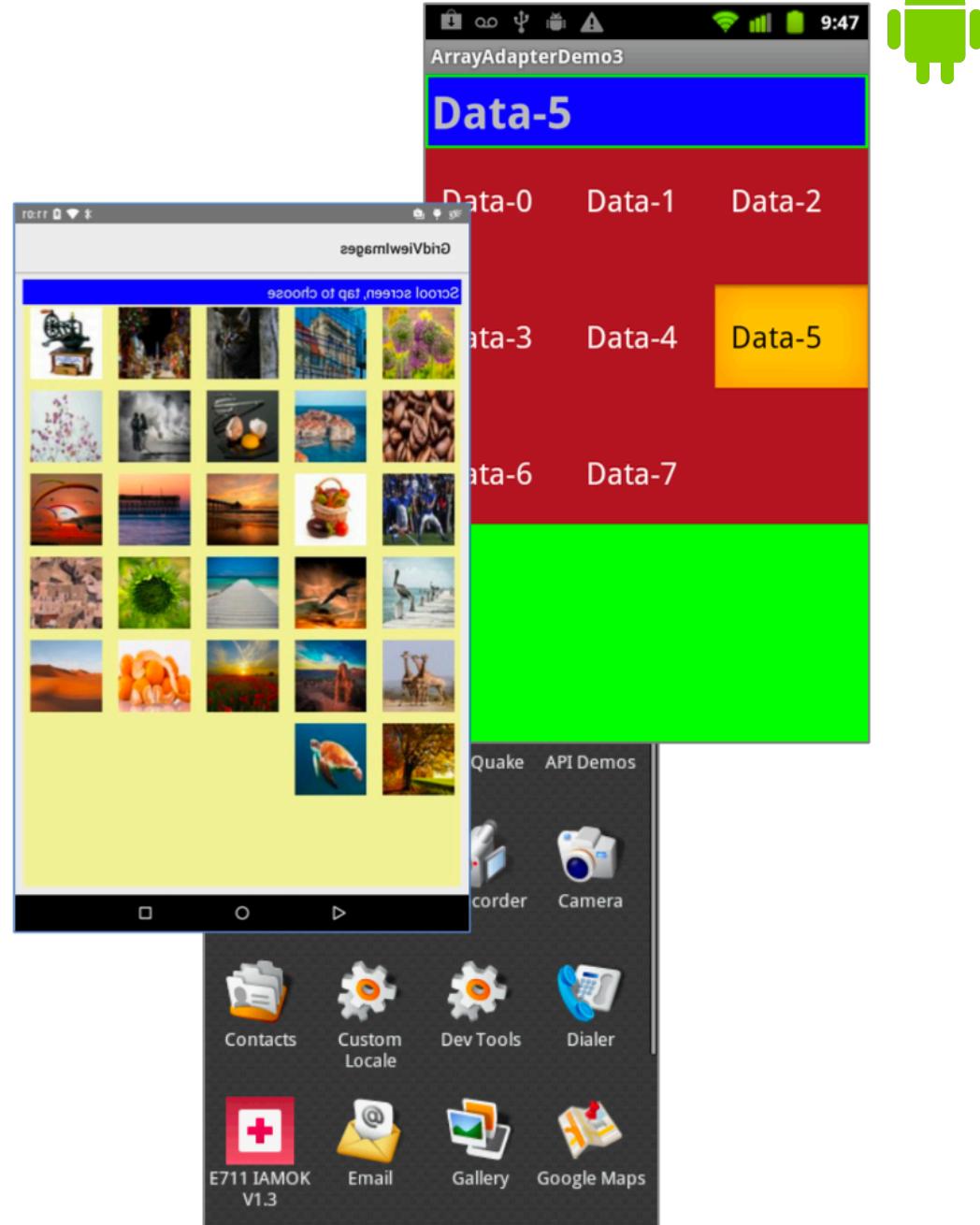
```
User user = users.get(0);  
adapter.remove(user);
```

- Xoá toàn bộ phần tử trong ListView

```
adapter.clear();
```

# GridView

- **GridView** là một **ViewGroup** dùng để trình bày dữ liệu dưới dạng một danh sách 2 chiều
- Dữ liệu trình bày trên **GridView** cũng được định dạng bởi **data adapter** giống **ListView**
- Mỗi **cell** của **GridView** có thể hiển thị hình ảnh, hoặc văn bản, hoặc cả 2





# GridView: Các thuộc tính hữu ích

---

- **android:numColumns**: cho biết số lượng cột của GridView. Có thể chọn “**auto\_fit**” để Android tự xác định số cột dựa trên không gian còn trống và các thuộc tính được liệt kê dưới đây.
- **android:verticalSpacing**, **android:horizontalSpacing**: khoảng cách theo chiều dọc và chiều ngang giữa các items/cells của GridView
- **android:columnWidth**: độ rộng của mỗi cột
- **android:stretchMode**: chỉ ra cách điều chỉnh kích thước cách thành phần của GridView khi có không gian trống



# GridView - stretchMode

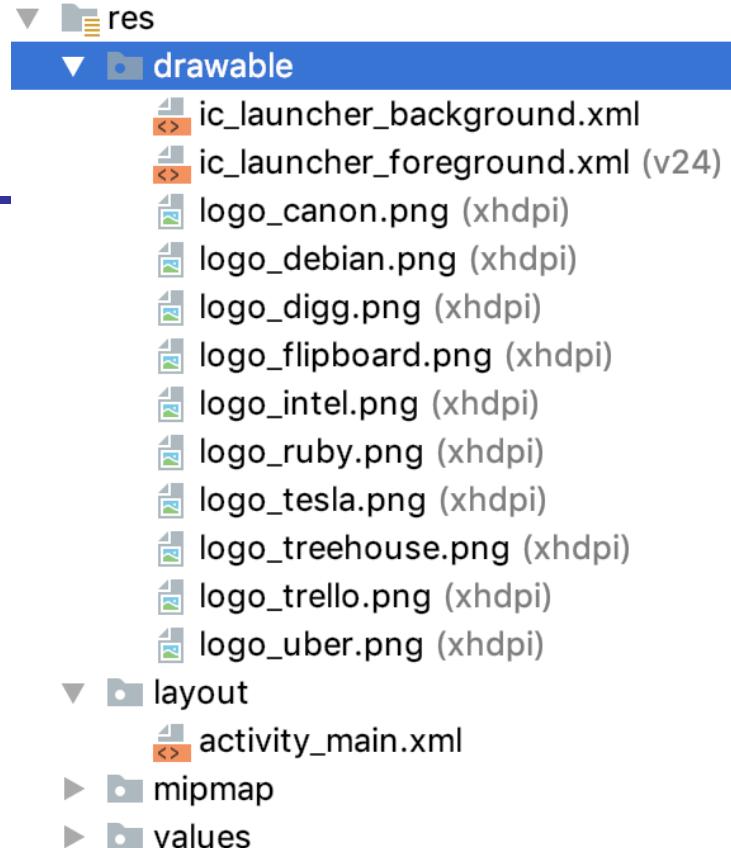
---

- Giả sử màn hình có chiều rộng là **320dp**, GridView được thiết lập:
  - `android:columnWidth="100dp"`
  - `android:horizontalSpacing="5dp"`
- → người dùng thấy GridView có 3 cột, chiếm **310dp** (3 cột x 100dp + 2 khoảng cách giữa các cột x 5dp)
- Nếu `android:stretchMode="columnWidth"`, 3 cột sẽ mở rộng thêm 3-4 dp mỗi cột để chiếm 10dp (=320-310) còn lại
- Nếu `android:stretchMode="spacingWidth"`, 2 khoảng cách giữa các cột sẽ mở rộng thêm 5dp mỗi khoảng cách để chiếm 10dp còn lại

# GridView: Demo

- Lặp lại các bước tương tự với ListView
- **Bước 1: Định nghĩa Model (File Company.java)**

```
public class Company {  
    String name;  
    int logo;  
  
    public Company(String name, int logo) {  
        this.name = name;  
        this.logo = logo;  
    }  
  
    public static ArrayList<Company> getMockData() {  
        ArrayList<Company> companies = new ArrayList<>();  
        String[] names = {"Apple", "Canon", "Debian", "Digg", "Flipboard", "Intel", "Ruby", "Tesla", "Treehouse",  
        "Trello", "Uber"};  
        int[] logos = {R.drawable.logo_apple, R.drawable.logo_canon, R.drawable.logo_debian, R.drawable.logo_digg,  
        R.drawable.logo_flipboard, R.drawable.logo_intel, R.drawable.logo_ruby, R.drawable.logo_tesla,  
        R.drawable.logo_treehouse, R.drawable.logo_trello, R.drawable.logo_uber};  
        for (int i = 0; i < names.length; i++) {  
            Company company = new Company(names[i], logos[i]);  
            companies.add(company);  
        }  
        return companies;  
    }  
}
```





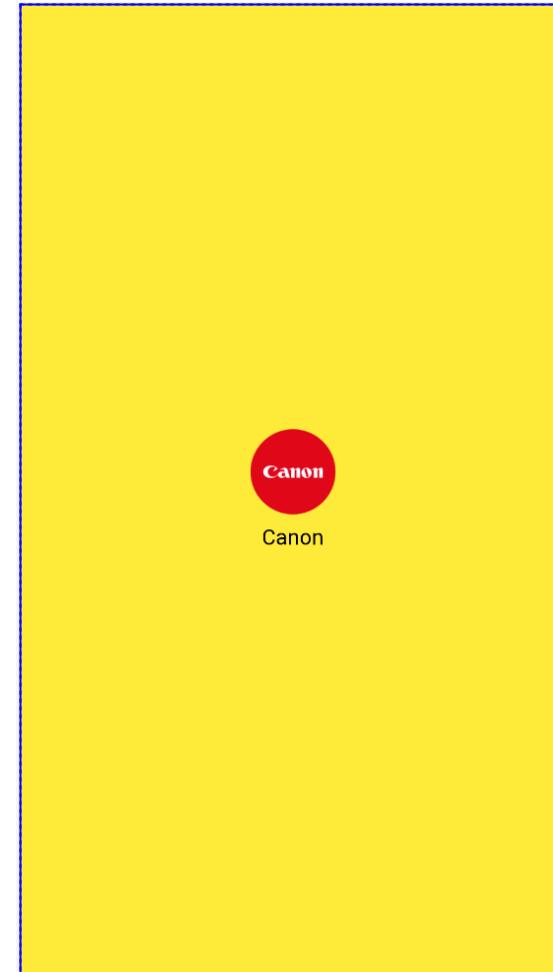
# GridView

- **Bước 2: Định nghĩa XML layout template cho 1 cell (file company\_item.xml)**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:padding="6dp"
        android:background="#FFEB3B"
        android:orientation="vertical">

    <ImageView android:id="@+id/ivLogo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/logo_canon" />

    <TextView android:id="@+id/tvName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="6dp"
        android:gravity="center"
        android:text="Canon"
        android:textColor="#000000"
        android:textSize="16sp" />
</LinearLayout>
```





# GridView

---

- **Bước 3: Định nghĩa Adapter (File CompanyAdapter.java)**

```
public class CompanyAdapter extends ArrayAdapter<Company> {  
    CompanyAdapter(Context context, ArrayList<Company> companies) {  
        super(context, R.layout.company_item, companies);  
    }  
  
    @Override  
    public View getView(int position, View convertView, ViewGroup parent) {  
        Company company = getItem(position);  
        ViewHolder viewHolder;  
        if (convertView == null) {  
            viewHolder = new ViewHolder();  
            convertView = LayoutInflater.from(getContext()).inflate(R.layout.company_item, parent, false);  
            viewHolder.logo = convertView.findViewById(R.id.ivLogo);  
            viewHolder.name = convertView.findViewById(R.id.tvName);  
            convertView.setTag(viewHolder);  
        } else {  
            viewHolder = (ViewHolder) convertView.getTag();  
        }  
        viewHolder.logo.setImageResource(company.logo);  
        viewHolder.name.setText(company.name);  
        return convertView;  
    }  
  
    private static class ViewHolder {  
        ImageView logo;  
        TextView name;  
    }  
}
```



# GridView

- **Bước 4:** Khai báo GridView (File activity\_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <GridView
        android:id="@+id/gvCompanies"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:columnWidth="100dp"
        android:horizontalSpacing="5dp"
        android:numColumns="auto_fit"
        android:stretchMode="spacingWidth"
        android:background="#03A9F4"
        android:verticalSpacing="5dp" />
</LinearLayout>
```

Item 1 Sub Item 1	Item 2 Sub Item 2	Item 3 Sub Item 3
Item 4 Sub Item 4	Item 5 Sub Item 5	Item 6 Sub Item 6
Item 7 Sub Item 7	Item 8 Sub Item 8	Item 9 Sub Item 9
Item 10 Sub Item 10	Item 11 Sub Item 11	Item 12 Sub Item 12
Item 13 Sub Item 13	Item 14 Sub Item 14	Item 15 Sub Item 15
Item 16 Sub Item 16	Item 17 Sub Item 17	Item 18 Sub Item 18
Item 19 Sub Item 19	Item 20 Sub Item 20	Item 21 Sub Item 21
Item 22 Sub Item 22	Item 23 Sub Item 23	Item 24 Sub Item 24

**Lưu ý:** các giá trị kích thước cần đưa vào res/values/dimens.xml

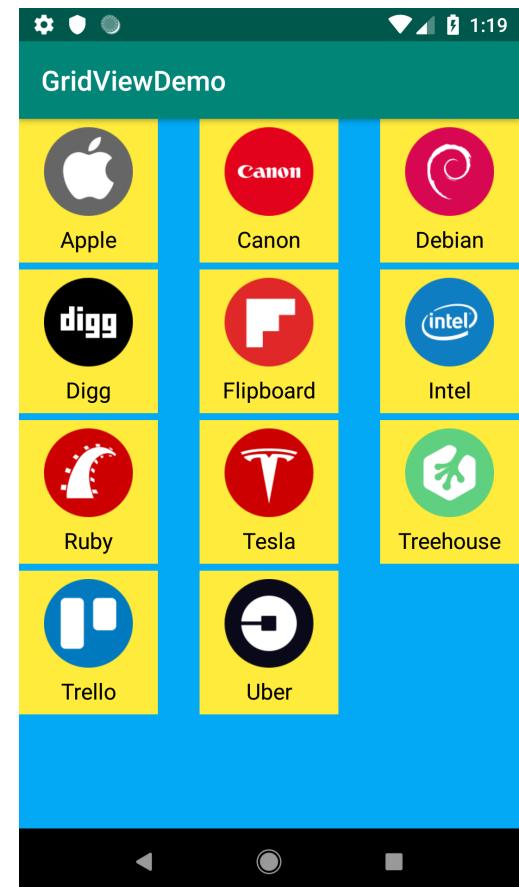




# GridView

## o Bước 5: Kết nối adapter với GridView

```
public class MainActivity extends AppCompatActivity {  
    ArrayList<Company> companies = Company.getMockData();  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        CompanyAdapter adapter = new CompanyAdapter(this, companies);  
        GridView gridView = (GridView) findViewById(R.id.gvCompanies);  
        gridView.setAdapter(adapter);  
    }  
}
```





# GridView

---

- **Bước 6:** Xử lý sự kiện chọn 1 cell của GridView
  - Cho MainActivity *implements* interface **AdapterView.OnItemClickListener** giúp MainActivity có khả năng lắng nghe sự kiện người dùng chọn 1 cell trên GridView
  - → Bắt buộc *override* phương thức **onItemClick(AdapterView<?> parent, View view, int position, long id)** tương ứng. Phương thức này được gọi khi người dùng chọn 1 cell trên GridView
  - Tham số position của hàm **onItemClick** cho biết index của cell được người dùng chọn

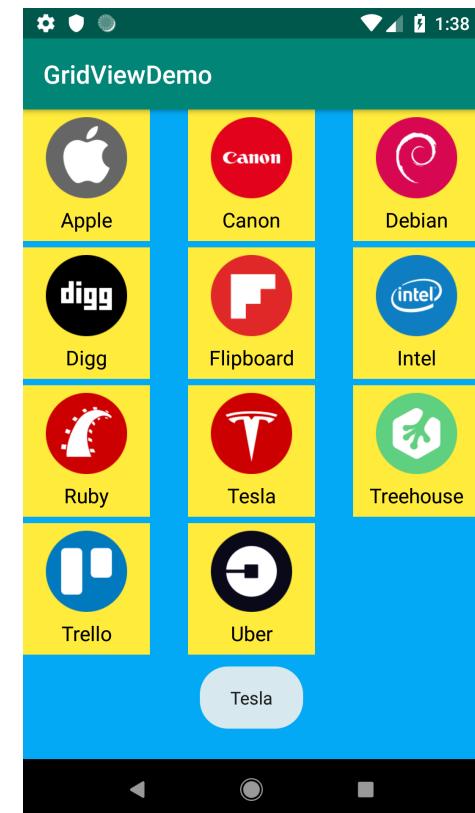
```
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    //...
}
```



# GridView

- **Bước 6** (tiếp theo): gắn đối tượng xử lý sự kiện **onItemClickListener** cho GridView chính là MainActivity

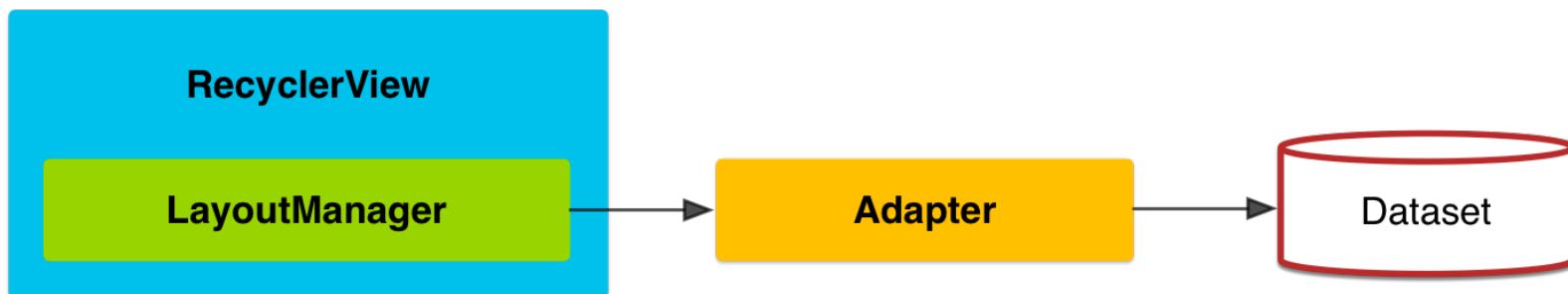
```
public class MainActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {  
    ArrayList<Company> companies;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        companies = Company.getMockData();  
        CompanyAdapter adapter = new CompanyAdapter(this, companies);  
        GridView gridView = (GridView) findViewById(R.id.gvCompanies);  
        gridView.setAdapter(adapter);  
        → gridView.setOnItemClickListener(this);  
    }  
  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        Toast.makeText(this, companies.get(position).name, Toast.LENGTH_SHORT).show();  
    }  
}
```





# RecyclerView

- Là một ViewGroup mới có thể thay thế cho ListView, GridView, nằm trong gói **library support v7**
- Khi sử dụng RecyclerView, sẽ cần làm việc với:
  - **RecyclerView.Adapter** – để xử lý danh sách dữ liệu và gắn dữ liệu vào từng row/cell tương ứng
  - **LayoutManager** – giúp bố cục vị trí các row/cell
  - **ItemAnimator** – giúp tạo các hiệu ứng khi thêm/xoá 1 row/cell





# RecyclerView – So sánh với ListView

---

- Bắt buộc có ViewHolder**  
(RecyclerView.ViewHolder)
- Tùy chỉnh layout theo chiều dọc hoặc ngang** dùng RecyclerView.LayoutManager
- Dễ dàng tạo hiệu ứng chọn/thêm/xoá** phần tử dùng RecyclerView.ItemAnimator
- Cần tuỳ biến data source** nhiều hơn dùng RecyclerView.Adapter
- Sử dụng RecyclerView.ItemDecoration** để tạo khoảng cách giữa các row và một số tùy chỉnh khác
- Chỉ có** RecyclerView.OnItemTouchListener để quản lý các touch event, không có sẵn interface giúp quản lý click event
- Không yêu cầu ViewHolder**
- Chỉ có layout theo chiều dọc (vertical)**
- Khó khăn** trong việc tạo animation
- Dễ quản lý** data source với ArrayAdapter...
- Sử dụng** thuộc tính android:divider để tạo khoảng cách giữa các row view
- Có** AdapterView.OnItemClickListener interface giúp quản lý sự kiện click trên các row



# RecyclerView - Demo

---

- Để sử dụng **RecyclerView**, cần thực hiện các bước sau:
  1. Khai báo thư viện trong **Gradle build file**
  2. Định nghĩa **model** class tương ứng với data source
  3. Thêm **RecyclerView** vào activity
  4. Tạo **XML layout template** tương ứng cho 1 **row/cell**
  5. Tạo **RecyclerView.Adapter** và **ViewHolder** để trình bày (render) 1 **row/cell**
  6. Gắn adapter tương ứng với data source vào **RecyclerView**



# RecyclerView - Demo

---

- Khai báo sử dụng thư viện trong Gradle build file “app/build.gradle”

```
dependencies {  
    // ...  
    implementation 'com.android.support:recyclerview-v7:28.0.0'  
}
```

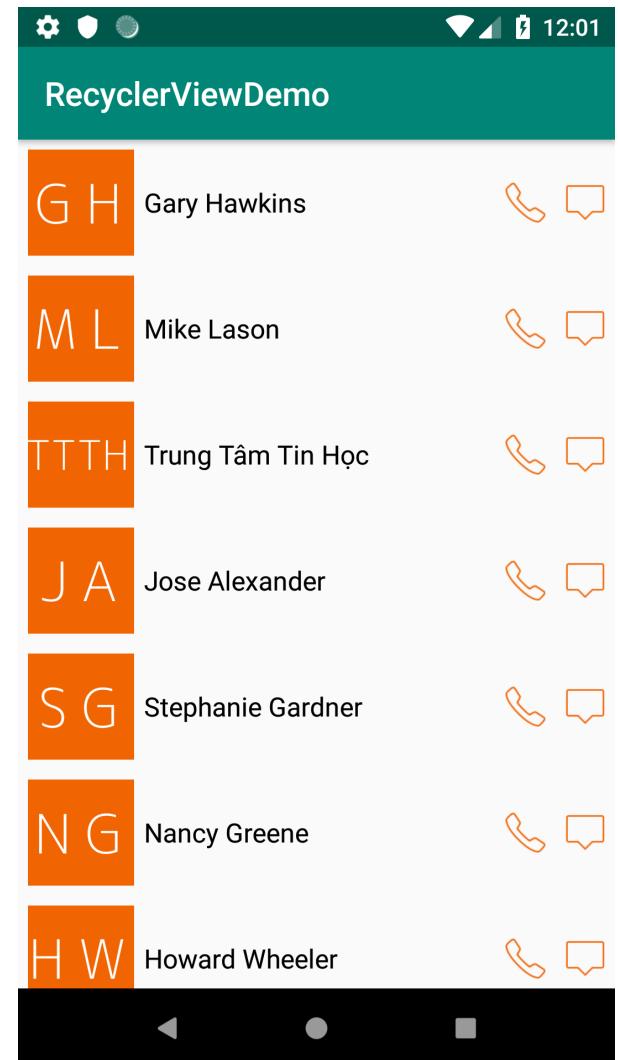
- Click “Sync Now” download để thư viện và tài nguyên tương ứng



# RecyclerView - Demo

- Định nghĩa model class: **Contact** (danh bạ) trong ví dụ này

```
public class Contact {  
    private String name;  
    private int avatar;  
    private String phone;  
  
    public Contact(String name, int avatar, String phone) {  
        this.name = name;  
        this.avatar = avatar;  
        this.phone = phone;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getPhone() {  
        return phone;  
    }  
  
    public int getAvatar() {  
        return avatar;  
    }  
}
```





# RecyclerView - Demo

- Định nghĩa **model** class - Có thể tạo dữ liệu mock up như sau

```
public class Contact {  
    // ...  
    public static ArrayList<Contact> getMockData() {  
        String[] names = {"Gary Hawkins", "Mike Lason", "Trung Tâm Tin Học", "Jose Alexander", "Stephanie  
        Gardner", "Nancy Greene", "Howard Wheeler", "Patricia Stephens", "Carl Garza", "Kelly Evans"};  
        int[] avatars = {R.drawable.img_01, R.drawable.img_02, R.drawable.img_03, R.drawable.img_04,  
        R.drawable.img_05, R.drawable.img_06, R.drawable.img_07, R.drawable.img_08, R.drawable.img_09,  
        R.drawable.img_10};  
        String[] phones = {"2701876007", "1671898312", "8308954038", "3508320983", "3317970165",  
        "5076902404", "6322118141", "3263941740", "5544270757", "5540715128"};  
        ArrayList<Contact> contacts = new ArrayList<>();  
        for (int i = 0; i < names.length; i++)  
            contacts.add(new Contact(names[i], avatars[i], phones[i]));  
        return contacts;  
    }  
}
```





# RecyclerView - Demo

---

- Khai báo **RecyclerView** trong XML layout file ([activity\\_main.xml](#))

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rvContacts"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```



# RecyclerView - Demo

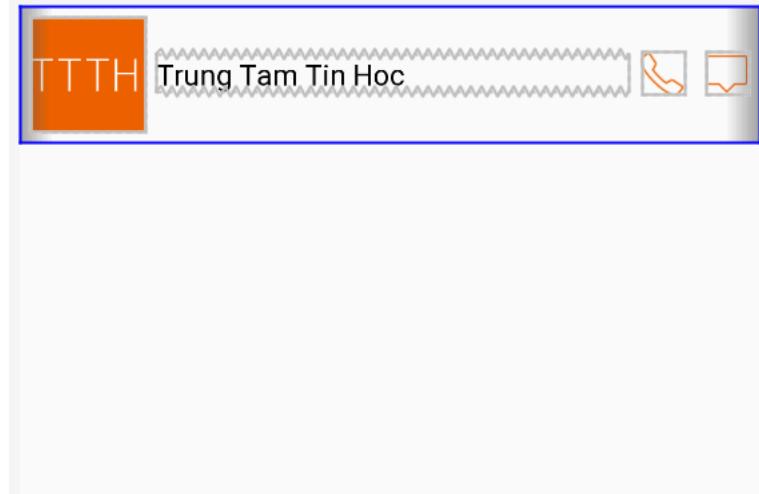
- Tạo XML layout template tương ứng với 1 row trong danh sách

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:orientation="horizontal"
    android:padding="@dimen/smallSpace">

    <ImageView
        android:id="@+id/ivAvatar"
        android:layout_width="@dimen/avatarSize"
        android:layout_height="@dimen/avatarSize"
        android:src="@drawable/img_03" />

    <TextView
        android:id="@+id/tvName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/smallSpace"
        android:layout_marginRight="@dimen/smallSpace"
        android:layout_weight="1"
        android:text="Trung Tam Tin Hoc"
        android:textColor="@color/black"
        android:textSize="@dimen/mediumFontSize" />

```



```
<ImageView
    android:id="@+id/ivCall"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="@dimen/smallSpace"
    android:src="@drawable/ic_call" />

    <ImageView
        android:id="@+id/ivText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="@dimen/smallSpace"
        android:src="@drawable/ic_text" />
</LinearLayout>
```



# RecyclerView - Demo

---

- File `res/values/dimens.xml`

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
    <dimen name="smallSpace">6dp</dimen>  
    <dimen name="avatarSize">64dp</dimen>  
    <dimen name="mediumFontSize">16sp</dimen>  
</resources>
```



# RecyclerView - Demo

---

- Tạo **RecyclerView.Adapter**: Thêm class mới **ContactAdapter** cùng với class **ViewHolder** như sau:

```
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
  
    public class ViewHolder extends RecyclerView.ViewHolder {  
        TextView tvName;  
        ImageView ivAvatar, ivCall, ivText;  
  
        public ViewHolder(View itemView) {  
            super(itemView);  
  
            tvName = itemView.findViewById(R.id.tvName);  
            ivAvatar = itemView.findViewById(R.id.ivAvatar);  
            ivCall = itemView.findViewById(R.id.ivCall);  
            ivText = itemView.findViewById(R.id.ivText);  
        }  
    }  
}
```



# RecyclerView - Demo

---

- Tạo **RecyclerView.Adapter** (2)

```
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
  
    // Định nghĩa ViewHolder  
  
    // Khai báo đối tượng chứa danh sách dữ liệu  
    private List<Contact> mContacts;  
    private Context mContext;  
  
    // Phương thức khởi tạo  
    public ContactAdapter(Context context, List<Contact> contacts) {  
        mContext = context;  
        mContacts = contacts;  
    }  
}
```



# RecyclerView - Demo

- Tạo **RecyclerView.Adapter** (3) – Override 3 phương thức chính cho adapter:
  - **onCreateViewHolder**: tạo layout cho mỗi mỗi row/cell và view holder tương ứng
  - **onBindViewHolder**: gán giá trị cho row/cell view dựa trên dữ liệu tại vị trí tương ứng (position)
  - **getItemCount**: xác định số lượng phần tử (row/cell) sẽ hiển thị lên RecyclerView

```
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
    // ...  
  
    @Override  
    public int getItemCount() {  
        return mContacts.size(); // trả về toàn bộ danh sách các contact  
    }  
}
```





# RecyclerView - Demo

- Tạo **RecyclerView.Adapter** (3) – Override 3 phương thức chính cho adapter:

```
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
    // ...  
    @NotNull  
    @Override  
    public ViewHolder onCreateViewHolder(@NotNull ViewGroup parent, int viewType) {  
        Context context = parent.getContext();  
        LayoutInflater inflater = LayoutInflater.from(context);  
        View contactView = inflater.inflate(R.layout.item_contact, parent, false);  
        ViewHolder viewHolder = new ViewHolder(contactView);  
        return viewHolder;  
    }  
  
    @Override  
    public void onBindViewHolder(@NotNull ViewHolder viewHolder, int position) {  
        Contact contact = mContacts.get(position);  
        viewHolder.tvName.setText(contact.getName());  
        viewHolder.ivAvatar.setImageResource(contact.getAvatar());  
    }  
}
```

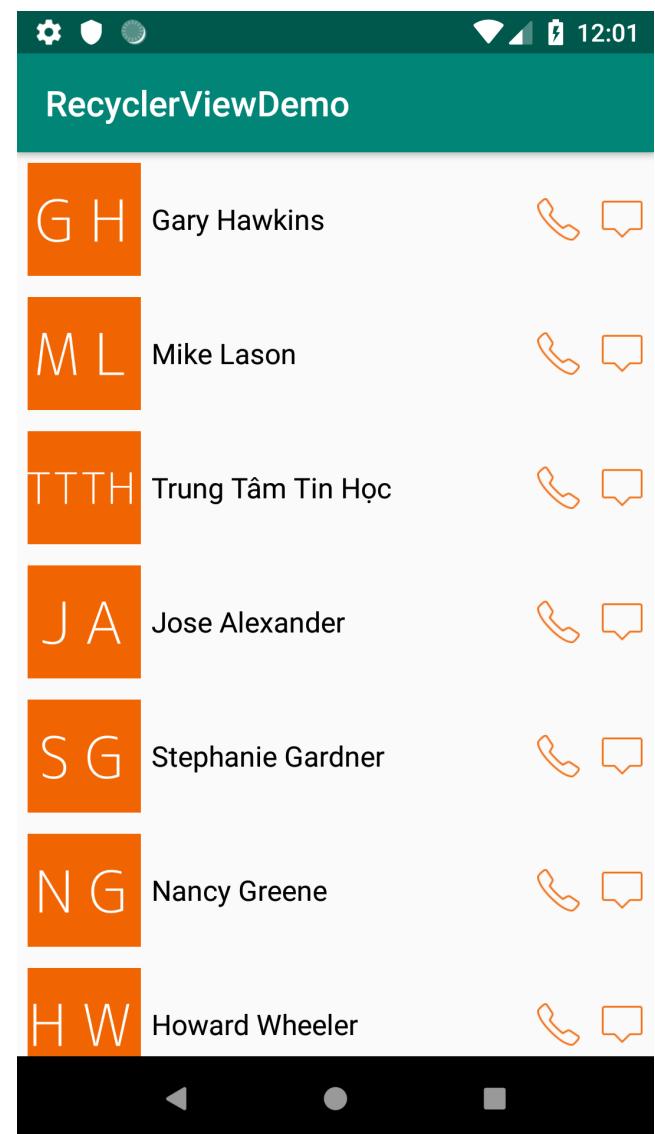




# RecyclerView - Demo

- Gắn Adapter vào RecyclerView

```
public class MainActivity extends AppCompatActivity {  
    ArrayList<Contact> contacts;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        RecyclerView rvContacts = findViewById(R.id.rvContacts);  
        contacts = Contact.getMockData();  
        ContactAdapter adapter = new ContactAdapter(this, contacts);  
        rvContacts.setAdapter(adapter);  
        // Gán layout manager cho RecyclerView  
        // LinearLayout sẽ tạo ra UI tương đương ListView  
        rvContacts.setLayoutManager(new LinearLayoutManager(this));  
    }  
}
```





# RecyclerView - Demo

- Thông báo để Adapter cập nhật lại RecyclerView khi có sự thay đổi về dữ liệu:
    - ListView hỗ trợ thêm/xoá dữ liệu trực tiếp, RecyclerView thì không  
→ khi có sự thay đổi về dữ liệu, cần thông báo cho Adapter để cập nhật render RecyclerView
    - Ví dụ 1: Thay đổi toàn bộ dữ liệu mới
- ```
contacts.clear(); contacts.addAll(Contact.getMockData()); adapter.notifyDataSetChanged();
```
- Ví dụ 2: Thêm 1 contact mới vào vị trí index = 0
- ```
contacts.add(0, new Contact("Tom C", R.drawable.img_11, "2701876008")); adapter.notifyItemInserted();
```
- Ví dụ 3: Xoá 1 contact tại vị trí index = 2
- ```
contacts.remove(2); adapter.notifyItemRemoved(2);
```
- Ví dụ 4: Thay đổi nội dung 1 contact tại vị trí index = 4
- ```
contacts.get(4).setPhone("2701876009"); adapter.notifyItemChanged(4);
```
- Ví dụ 5: Thêm 3 contact mới
- ```
ArrayList<Contact> newItems = Contact.get3Contacts(); int currentSize = adapter.getItemCount(); contacts.addAll(newItems); adapter.notifyItemRangeInserted(currentSize, newItems.size());
```



# RecyclerView - Demo

- Xử lý sự kiện click lên 2 icon “call” và “text” – Cách 1

@Override

```
public void onBindViewHolder(@NonNull ViewHolder viewHolder, int position) {
```

```
    Contact contact = mContacts.get(position);
```

```
    viewHolder.tvName.setText(contact.getName());
```

```
    viewHolder.ivAvatar.setImageResource(contact.getAvatar());
```

```
    viewHolder.ivCall.setOnClickListener(new View.OnClickListener() {
```

@Override

```
        public void onClick(View v) {
```

```
            Toast.makeText(mContext, "Call " + contact.getPhone(), Toast.LENGTH_SHORT).show();
```

```
        }
```

```
    });
```

```
    viewHolder.ivText.setOnClickListener(new View.OnClickListener() {
```

@Override

```
        public void onClick(View v) {
```

```
            Toast.makeText(mContext, "Text " + contact.getPhone(), Toast.LENGTH_SHORT).show();
```

```
        }
```

```
    });
```

```
}
```





# RecyclerView - Demo

- Xử lý sự kiện click lên 2 icon “call” và “text” – Cách 2

```
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
    // ...  
    public interface OnContactClickListener {  
        void onContactCallClick(Contact contact);  
        void onContactTextClick(Contact contact);  
    }  
  
    private List<Contact> mContacts;  
    private OnContactClickListener mListener;  
  
    public ContactAdapter(List<Contact> contacts, OnContactClickListener listener) {  
        mContacts = contacts;  
        mListener = listener;  
    }  
    // ...  
}
```



# RecyclerView - Demo

- Xử lý sự kiện click lên 2 icon “call” và “text” – Cách 2

```
public class ContactAdapter extends RecyclerView.Adapter<ContactAdapter.ViewHolder> {  
    // ...  
    @Override  
    public void onBindViewHolder(@NonNull ViewHolder viewHolder, int position) {  
        final Contact contact = mContacts.get(position);  
        viewHolder.tvName.setText(contact.getName());  
        viewHolder.ivAvatar.setImageResource(contact.getAvatar());  
        viewHolder.ivCall.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                mListener.onContactCallClick(contact);  
            }  
        });  
        viewHolder.ivText.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                mListener.onContactTextClick(contact);  
            }  
        });  
    }  
    // ...  
}
```



# RecyclerView - Demo

- Xử lý sự kiện click lên 2 icon “call” và “text” – Cách 2

```
public class MainActivity extends AppCompatActivity implements ContactAdapter.OnContactClickListener {  
    ArrayList<Contact> contacts;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // ...  
        ContactAdapter adapter = new ContactAdapter(contacts, this);  
        rvContacts.setAdapter(adapter);  
        rvContacts.setLayoutManager(new LinearLayoutManager(this));  
    }  
  
    @Override  
    public void onContactCallClick(Contact contact) {  
        Toast.makeText(this, "Call " + contact.getPhone(), Toast.LENGTH_SHORT).show();  
    }  
  
    @Override  
    public void onContactTextClick(Contact contact) {  
        Toast.makeText(this, "Text " + contact.getPhone(), Toast.LENGTH_SHORT).show();  
    }  
}
```



# Spinner

- Tương tự như DropDownList trong các môi trường khác
- Nguyên lý tương tự ListView
- Demo:





# Spinner - Demo

---

- o activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <android.support.v7.widget.AppCompatSpinner
        android:id="@+id/spinnerCar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ImageView
        android:id="@+id/ivCar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="10dp" />

</LinearLayout>
```



# Spinner - Demo

```
public class MainActivity extends AppCompatActivity {  
    AppCompatSpinner spinnerCar;  
    ImageView ivCar;  
    String[] cars = {"Audi R8", "BMW i8", "Tesla Model S"};  
    int[] imgs = {R.drawable.audi_r8, R.drawable.bmw_i8, R.drawable.tesla};  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        spinnerCar = findViewById(R.id.spinnerCar);  
        ivCar = findViewById(R.id.ivCar);  
  
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, cars);  
        spinnerCar.setAdapter(adapter);  
    }  
}
```

Sử dụng ArrayAdapter với template mẫu cho mỗi dòng  
của Spinner là mặc định của hệ thống



# Spinner - Demo

---

- Xử lý sự kiện chọn (select) 1 item của Spinner

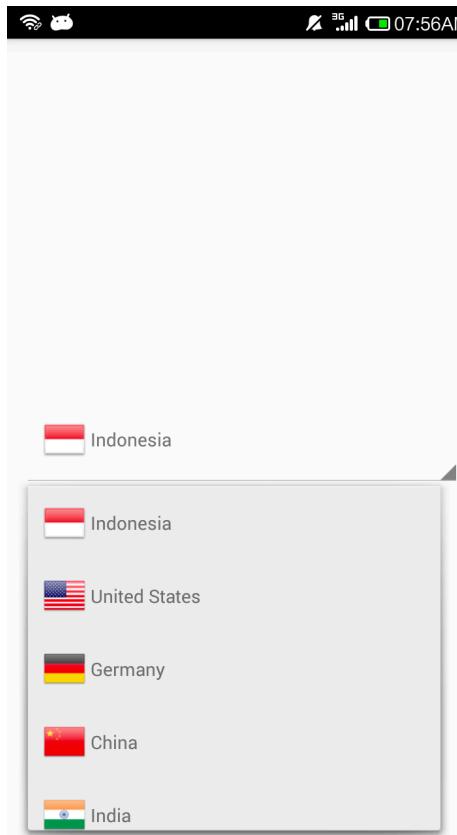
```
public class MainActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {  
    // ...  
    int[] imgs = {R.drawable.audi_r8, R.drawable.bmw_i8, R.drawable.tesla};  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        // ...  
  
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.simple_spinner_dropdown_item, cars);  
        spinnerCar.setAdapter(adapter);  
        spinnerCar.setOnItemSelectedListener(this);  
    }  
  
    @Override  
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {  
        ivCar.setImageResource(imgs[position]);  
    }  
  
    @Override  
    public void onNothingSelected(AdapterView<?> parent) {  
    }  
}
```





# Spinner – Tuỳ chỉnh

- Có thể tùy chỉnh spinner thay vì dùng layout template cho 1 dòng là của hệ thống, bằng cách:
  - Định nghĩa layout template cho 1 dòng tùy theo yêu cầu bài toán
  - Định nghĩa adapter cho spinner, kế thừa từ **ArrayAdapter** (tương tự ListView)





## Q&A

---

