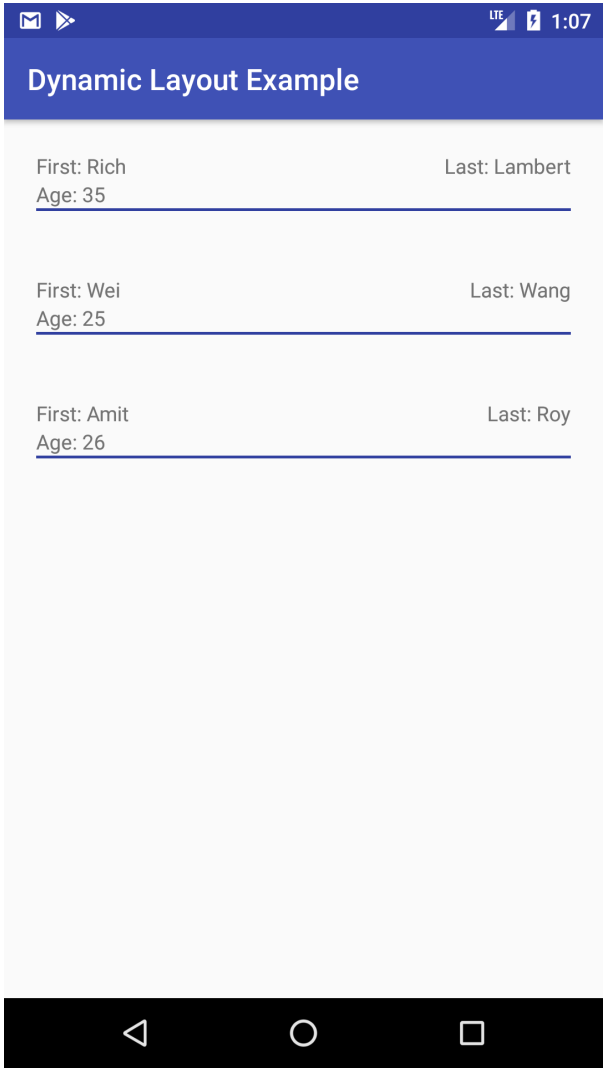
How to Create a Dynamic Layout and Inflate it?

Objectives

- Create a scrollable list of items without using ListView/RecyclerView.
- Create the list.
  - Design a template layout for the items.
  - Create inflated instances of the items.
  - Add them to a container.
  - Add the container to the ScrollView.

Example

- Create and display a list of users in an ArrayList.
- User class:
  - First name.
  - Last name.
  - Age.
- ArrayList contains three users:
  - "Rich", "Lambert", 35
  - "Wei", "Wang", 25
  - "Amit", "Roy", 26



Step 1: creating the User Class for accommodating the users in our example

```
public class User {
    String firstName, lastName;
    int age;

    User(String firstName, String lastName, int age){
        this.firstName = firstName;
        this.lastName =lastName;
        this.age = age;
    }

    public String getFirstName() {
        return firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public int getAge() {
        return age;
    }
}
```

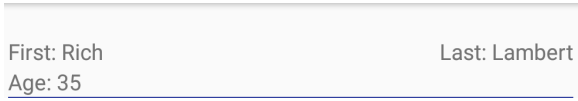
Step 2: createUsers class for generating three users

```
public class createUsers {
    public static ArrayList generate(){
        ArrayList<User> list = new ArrayList<User>();
        User user1 = new User("Rich", "Lambert", 35);
        User user2 = new User("Wei", "Wang", 25);
        User user3 = new User("Amit", "Roy", 26);
        list.add(user1);
        list.add(user2);
        list.add(user3);
        return list;
    }
}
```

Step 3: create a common layout file for each list item (listiem.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="22sp">

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/tv1"
            android:gravity="start"/>
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/tv2"
            android:gravity="end"/>
    </LinearLayout>

    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/tv3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:gravity="left"/>
    </LinearLayout>

</LinearLayout>
```

Two horizontal linear layouts for creating the list items as we displayed in the example



Step 4: create a UI class (listitemUI) extending LinearLayout to represent the listitem.xml

```
public class listItemUI extends LinearLayout{

    public TextView first, last, age;

    public listItemUI(Context context) {
        super(context);

    }
}
```

1. Three TextViews for three TextViews in listitem.xml
2. Constructor for the Class. Pass the context as a parameter.
3. We need to write a method to inflate the layout of listitem.xml

Step 5: write inflateXML() method and call it from the constructor

```
public class listItemUI extends LinearLayout{

    public TextView first, last, age;

    public listItemUI(Context context) {
        super(context);
        inflateXML(context);
    }

    private void inflateXML(Context context) {
        LayoutInflater inflater =
          (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View itemView = inflater.inflate(R.layout.listitem, this);
        this.first = (TextView) findViewById(R.id.tv1);
        this.last = (TextView) findViewById(R.id.tv2);
        this.age = (TextView) findViewById(R.id.tv3);
    }
}
```

Step 5: write inflateXML() method and call it from the constructor.

```
public class listItemUI extends LinearLayout{

    public TextView first, last, age;

    public listItemUI(Context context) {
        super(context);
        inflateXML(context);
    }

    private void inflateXML(Context context) {
        LayoutInflater inflater =
          (LayoutInflater)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View itemView = inflater.inflate(R.layout.listitem, this);
        this.first = (TextView) findViewById(R.id.tv1);
        this.last = (TextView) findViewById(R.id.tv2);
        this.age = (TextView) findViewById(R.id.tv3);
    }
}
```

1. It inflates the layout of listitem.xml in the context you pass.
2. Then set the TextViews to the appropriate TextViews from listitem.xml

Step 6: Main Activity

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ScrollView sv_main = (ScrollView) findViewById(R.id.svmain);
        LinearLayout container = new LinearLayout(this);
        container.setOrientation(LinearLayout.VERTICAL);
        ArrayList<User> listOfUsers = new ArrayList<User>();

        listOfUsers = createUsers.generate();
    }
}
```

1. Create a ScrollView and set it to the ScrollView in main layout (activity_main.xml).
2. Create a container for the list of layout items. I created a LinearLayout with the Vertical orientation.
3. Created an ArrayList of three users.

Step 6: Main Activity

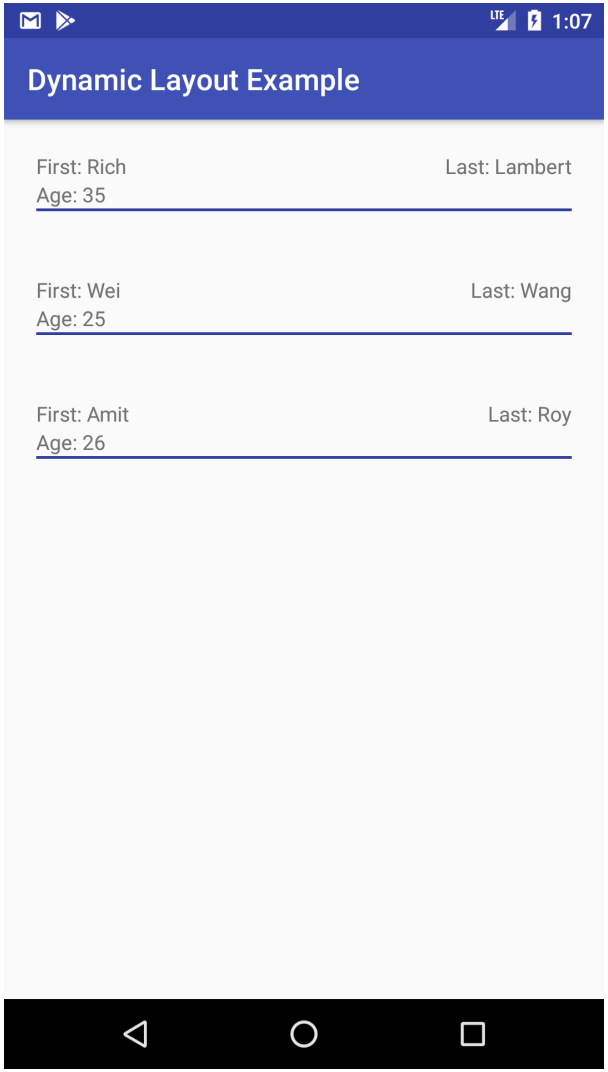
```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ScrollView sv_main = (ScrollView) findViewById(R.id.svmain);
        LinearLayout container = new LinearLayout(this);
        container.setOrientation(LinearLayout.VERTICAL);
        ArrayList<User> listOfUsers = new ArrayList<User>();

        listOfUsers = createUsers.generate();

        for (User user: listOfUsers){
            listItemUI item = new listItemUI(this);
            View itemView = (View)item;
            item.first.setText("First: "+user.firstName);
            item.last.setText("Last: "+user.lastName);
            item.age.setText("Age: "+user.age);

            container.addView(itemView);

        }

        sv_main.addView(container);

    }
}
```

Step 7: Construct the list

```
for (User user: listOfUsers){
            listItemUI item = new listItemUI(this);
            View itemView = (View)item;
            item.first.setText("First: "+user.firstName);
            item.last.setText("Last: "+user.lastName);
            item.age.setText("Age: "+user.age);
            container.addView(itemView);
}
sv_main.addView(container);
```

1. Create a listItemUI item and pass the current context to create new UI object.
2. Create a View for the item.
3. Set texts to TextViews.
4. Add the itemView in the container LinearLayout.
5. After adding all the users, add the container to the ScrollView.
6. You are done!

Output


1:07
Dynamic Layout Example
First: Rich
Last: Lambert
Age: 35
First: Wei
Last: Wang
Age: 25
First: Amit
Last: Roy
Age: 26