

Build X: Android

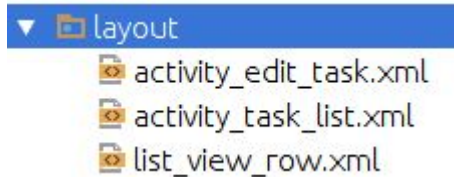
Lecture Six:
Putting it All Together

Notes: <http://android.kcl.tech/>



Recap

Recap: We've Made Layouts...



```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">
```

```
<ProgressBar
    android:id="@+id/loading_icon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="android:progressBarStyleInverse"/>
```

```
<ScrollView
    android:id="@+id/main_content"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:visibility="gone">
```

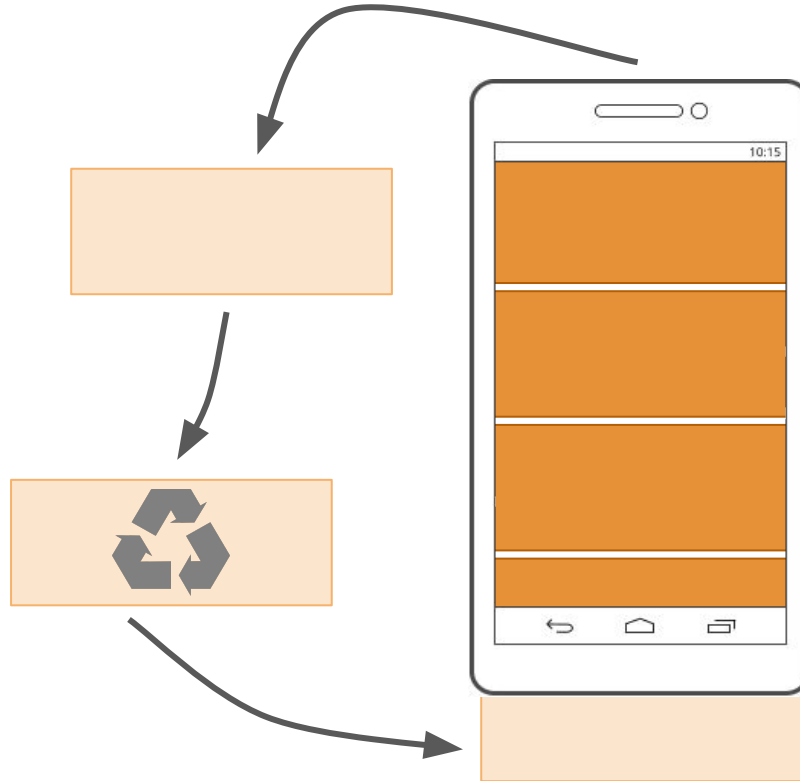
Recap: We Have Activities...

```
public class TaskListActivity extends AppCompatActivity {
```

```
public class EditTaskActivity extends AppCompatActivity {
```

```
<activity  
    android:name=".TaskListActivity"  
    android:label="@string/app_name">  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN" />  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>  
<activity android:name=".EditTaskActivity" />
```

Recap: We Can Use ListViews...



Recap: We Built a Task Model...

```
public class Task {  
    private int id = -1; // ID defaults to -1, meaning "no ID has been set yet"  
    private String title;  
    private String notes;  
    private DateTime dueDate;  
    private boolean isComplete;
```

Recap: We Made a Database...

```
public class DbHelper extends SQLiteOpenHelper {  
  
    public static final String DB_NAME = "DB";  
    public static final int DB_VERSION = 1;  
  
1   public DbHelper(Context context) { super(context, DB_NAME, null, DB_VERSION); }  
  
    @Override  
1   public void onCreate(SQLiteDatabase db) {...}  
  
    @Override  
1   public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {...}
```

Time to Put
It All Together!

Remaining Jobs

- Fetch tasks from the database and put them into a ListView
- Show task notes and options when a task is clicked
- Allow for completion of a task
- Add a “+” button to create a new task
- Save a task in the database when the form is filled out
- Allow for editing of a task

Fair Warning:

Lots of code coming up

Remaining Jobs

- **Fetch tasks from the database and put them into a ListView**
- Show task notes and options when a task is clicked
- Allow for completion of a task
- Add a “+” button to create a new task
- Save a task in the database when the form is filled out
- Allow for editing of a task



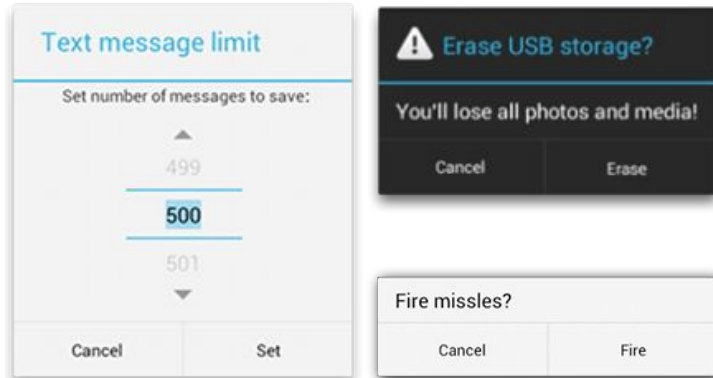
Coding Time

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- **Show task notes and options when a task is clicked**
- ~~Allow for completion of a task~~
- ~~Add a “+” button to create a new task~~
- ~~Save a task in the database when the form is filled out~~
- ~~Allow for editing of a task~~

Alert Dialogs

- An alert dialog is a “pop-up” on Android that can contain pretty much anything you want
 - Text, an image, input fields, progress bars, buttons, etc.



Alert Dialogs: Builders

- We could create one from scratch, but it's easier to use a **design pattern** called a **builder**.
- The **builder pattern** is used in many languages and work like a real builder: you tell them how you want a complex object to look/work, and they build it for you.

Alert Dialogs: Builders

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);  
  
builder.setOption( ... )  
    .setOption( ... )  
    .setOption( ... );  
  
AlertDialog dialog = builder.create();  
dialog.show();
```


Alert Dialogs: Builders

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>) ;
```

```
builder.setOption( ... )  
        .setOption( ... )  
        .setOption( ... );
```

```
AlertDialog dialog = builder.create();  
dialog.show();
```

Alert Dialogs: Builders

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);
```

```
builder.setOption( ... )  
        .setOption( ... )  
        .setOption( ... );
```

```
AlertDialog dialog = builder.create();  
dialog.show();
```

Alert Dialogs: Builders

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);
```

```
builder.setOption( ... )  
        .setOption( ... )  
        .setOption( ... );
```

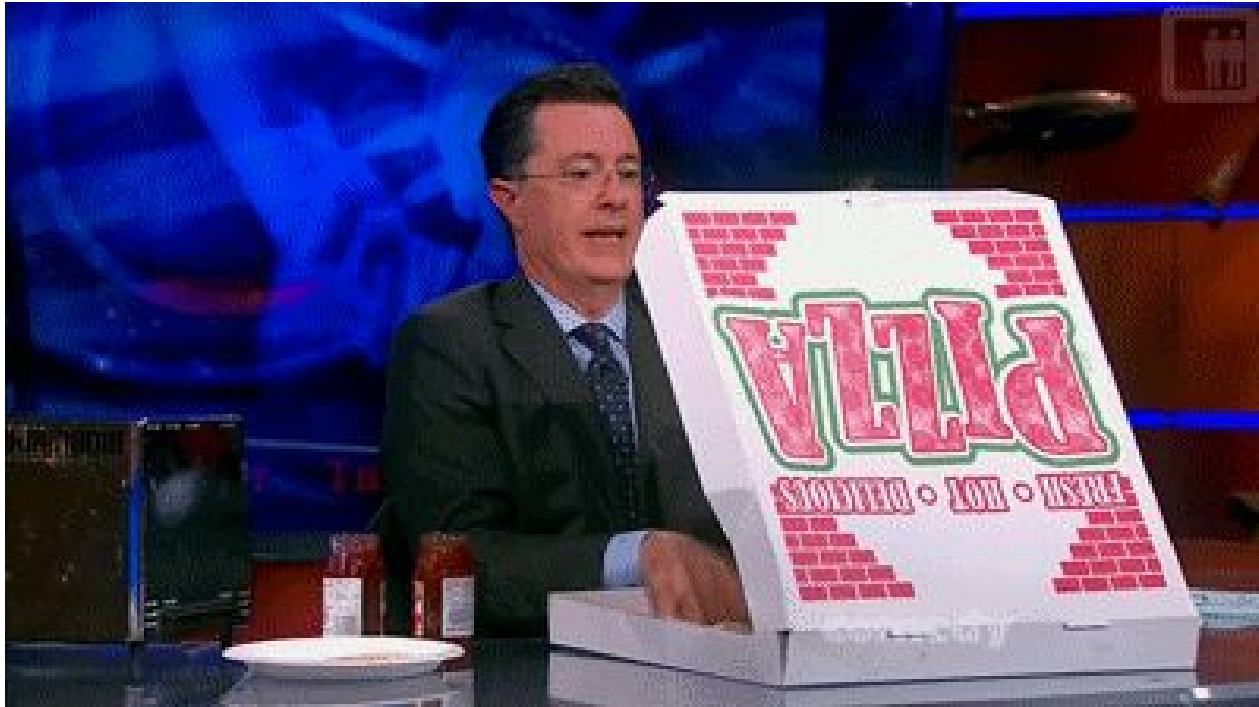
```
AlertDialog dialog = builder.create();  
dialog.show();
```

Alert Dialogs: Builders

```
AlertDialog.Builder builder =  
    new AlertDialog.Builder(<<activity ref>>);
```

```
builder.setOption( ... )  
        .setOption( ... )  
        .setOption( ... );
```

```
AlertDialog dialog = builder.create();  
dialog.show();
```



Coding Time

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- **Allow for completion of a task**
- Add a “+” button to create a new task
- Save a task in the database when the form is filled out
- Allow for editing of a task

Completing a Task

1. Find out which list item they clicked on.
2. Get the `Task` object from that position.
3. Set `isCompleted` to `true`.
4. Save it in the database again.



Coding Time







Break! (10 mins)

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- ~~Allow for completion of a task~~
- **Add a “+” button to create a new task**
- ~~Save a task in the database when the form is filled out~~
- ~~Allow for editing of a task~~



Coding Time

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- ~~Allow for completion of a task~~
- ~~Add a “+” button to create a new task~~
- **Save a task in the database when the form is filled out**
- Allow for editing of a task

Saving a Task

1. Get all the input the user gave.
2. Create a `Task` object.
3. Save it in the database.
4. Send the user back to the list.



Coding Time

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- ~~Allow for completion of a task~~
- ~~Add a “+” button to create a new task~~
- ~~Save a task in the database when the form is filled out~~
- **Allow for editing of a task**

Editing a Task

1. Send the user to the edit page, with the ID in the intent extras.
2. Load the relevant `Task` object and pre-fill the inputs.
3. Let the save button carry on as before!



Coding Time





Remaining Jobs

- **Fetch tasks from the database and put them into a ListView**
- Show task notes and options when a task is clicked
- Allow for completion of a task
- Add a “+” button to create a new task
- Save a task in the database when the form is filled out
- Allow for editing of a task

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- **Show task notes and options when a task is clicked**
- ~~Allow for completion of a task~~
- ~~Add a “+” button to create a new task~~
- ~~Save a task in the database when the form is filled out~~
- ~~Allow for editing of a task~~

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- **Allow for completion of a task**
- Add a “+” button to create a new task
- Save a task in the database when the form is filled out
- Allow for editing of a task

Remaining Jobs

- Fetch tasks from the database and put them into a ListView
- Show task notes and options when a task is clicked
- Allow for completion of a task
- **Add a “+” button to create a new task**
- Save a task in the database when the form is filled out
- Allow for editing of a task

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- ~~Allow for completion of a task~~
- ~~Add a “+” button to create a new task~~
- **Save a task in the database when the form is filled out**
- Allow for editing of a task

Remaining Jobs

- ~~Fetch tasks from the database and put them into a ListView~~
- ~~Show task notes and options when a task is clicked~~
- ~~Allow for completion of a task~~
- ~~Add a “+” button to create a new task~~
- ~~Save a task in the database when the form is filled out~~
- **Allow for editing of a task**

Remaining Jobs

- = ~~Fetch tasks from the database and put them into a ListView~~
- = ~~Show task notes and options when a task is clicked~~
- = ~~Allow for completion of a task~~
- = ~~Add a “+” button to create a new task~~
- = ~~Save a task in the database when the form is filled out~~
- = ~~Allow for editing of a task~~

A full-page background image featuring Leonardo DiCaprio in a tuxedo, smiling and holding a glass of champagne. He is surrounded by a festive party atmosphere with blurred lights and other guests in the background.

**You Just Built
Your Own App!**

So What Now?

- **Lecture 7: Networking & APIs**
 - Using other services to do some really cool stuff!
- **Lecture 8: To be confirmed**
 - Suggestions welcome!
- **Lecture 9: Open session**
 - A few new topics, problem solving and points of interest.

Basically...

Keep coming!