Android 101:

Build an app from start to finish

Putting it all together

The app variable demystified

So far we have seen events that happen per Activity

But, we can have events for the whole application.

GirlDevelopIt.java extends Application

Created when the Application is opened and saves data even when the application isn't in view

DataStore

First, let's explain DataStore.
A file in the Utils folder that stores and retrieves Data

```
private static final String PREFS_NAME =
"girlDevelopItPrefs";

public void saveToPrefs(String key, String value){

  SharedPreferences prefs =
  context.getSharedPreferences(PREFS_NAME,
  Context.MODE_PRIVATE);

  SharedPreferences.Editor editor = prefs.edit();

  editor.putString(key, value);

  editor.commit();
}
```

DataStore

First, let's explain DataStore.
A file in the Utils folder that stores and retrieves Data

```
public String getFromPrefs(String key, String
defaultValue){

  SharedPreferences prefs =
  context.getSharedPreferences(PREFS_NAME,
  Context.MODE_PRIVATE);

   return prefs.getString(key, defaultValue);
}
```

DataStore

First, let's explain DataStore.
A file in the Utils folder that stores and retrieves Data

```
public void saveExternalImageData(ArrayList<ImageModel>
imagesList){
    final File suspend_f=new File(app.getCacheDir(),
"images");
   ///lots of code here to save the ArrayList to an
   ///external File
}

public ArrayList<ImageModel> getExternalImageData(){
    ArrayList<ImageModel> imagesList = new
ArrayList<ImageModel>();
   final File suspend_f=new File(app.getCacheDir(),
"images");
    ///lots of code here to get an ArrayList from the
    /// external file
   return imagesList;
}
```

GirlDevelopIt

Saves and Retrieves Data needed throughout the lifetime of the application.

```
public class GirlDevelopIt extends Application {
    private DataStore store;

    @Override
    public void onCreate()
    {
        super.onCreate();
        store = new DataStore(this);
    }
```

Saves and Retrieves Data needed throughout the lifetime of the application.

```
public void setUsername(String name){
    store.saveToPrefs("USERNAME", name);
}

public String getUsername(){
    return store.getFromPrefs("USERNAME", "");
}
```

GirlDevelopIt

Saves and Retrieves Data needed throughout the lifetime of the application.

```
public void setImages(ArrayList<ImageModel> img){
   store.saveExternalImageData(img);
}

public ArrayList<ImageModel> getImages(){
  return store.getExternalImageData();
}
```

GalleryActivity

initElements()
Adding an onClickListener to our ListView

```
imageListView =
(ListView)this.findViewById(R.id.imageListView);

imageListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
  @Override
  public void onItemClick(AdapterView<?> arg0, View
  arg1, int position, long arg3) {

    Bundle b = new Bundle();
    b.putInt("image_position", position);

    Intent intent = new Intent(GalleryActivity.this,
ViewPictureActivity.class);
    intent.putExtras(b);

    startActivity(intent);
  }
});
```

AndroidManifest

Preparing for ViewPictureActivity

```
<activity
  android:name=".views.ViewPictureActivity"

  android:configChanges="keyboardHidden|orientation"

  android:screenOrientation="portrait"/>
```

viewpicture.xml

Build the layout for seeing a picture

"Taken By" TextView
Username TextView

"Taken On" TextView
Date TextView

"Description" TextView
Description TextView

ImageView that takes up the rest of the space

viewpicture.xml

Build the layout for seeing a picture

```
<LinearLayout  android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">

  <TextView android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Taken by:"
       android:textColor="@color/PURPLE"
       android:paddingRight="16dip"/>

  <TextView android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="username here"
       android:id="@+id/username"
       android:textColor="@color/BLACK"/>

</LinearLayout>
```

ViewPictureActivity

Declare Variables

```
private GirlDevelopIt app;
private TextView username;
private TextView date;
private TextView description;
private ImageView image;
private int imagePosition;
```

ViewPictureActivity

onCreate

```
super.onCreate(savedInstanceState);

setContentView(R.layout.viewpicture);

this.app = (GirlDevelopIt)getApplicationContext();

Bundle b;
b = getIntent().getExtras();
imagePosition = b.getInt("image_position");

initElements();
```

ViewPictureActivity

initElements()

```
username =
(TextView)this.findViewById(R.id.username);

date = (TextView)this.findViewById(R.id.date);

description =
(TextView)this.findViewById(R.id.description);


image = (ImageView)this.findViewById(R.id.image);
```

ViewPictureActivity

initElements()

```
ImageModel imageModel =
app.getImages().get(imagePosition);

username.setText(imageModel.getUsername());

SimpleDateFormat sdf = new SimpleDateFormat("MM dd,
yyyy HH:mm", Locale.getDefault());

Date d = new Date(imageModel.getDateCreated());
date.setText(sdf.format(d));

description.setText(imageModel.getDescription());
```

ViewPictureActivity

initElements()

```
String pathToImage = imageModel.getPathToImage();

File selFile=new File(pathToImage);

Bitmap thumbnailBmp = Utils.decodeFile(selFile);

image.setImageBitmap(thumbnailBmp);

image.setScaleType(ImageView.ScaleType.CENTER_INSIDE)
;
```

Resources

Principles of Android Mobile Design
Dealing with Images and Graphics on Android
Android Tools
Udemy Android

Questions?