15. Using Libraries

Libraries

- Many Android developers have produced useful **libraries**.There is a **Maven repository** to store various libraries.
- This makes it easy to add them to your Android Studio projects.
- Most libraries use permissive licenses so that you can use them for free and can include them in the code of commercial apps/products.

maven

Adding a library to your project

- Edit the **build.gradle** file for your 'app' module and add lines to the following section at the bottom.
  - You can usually find out what file name to write below by going to various libraries' home pages / GitHub pages.
- dependencies {

  ```
      implementation fileTree(dir: 'libs', include: ['*.jar’])
      testImplementation 'junit:junit:4.12’
      implementation 'com.android.support:appcompat-v7:23.1.1’
      implementation 'your library file here’
      implementation 'your library file here' ...
      implementation 'your library file here' }
  ```

Picasso library

- **Picasso** is a powerful library for manipulating images.
  - written by Square, inc.
  - http://square.github.io/picasso/

- To add Picasso to your project:

```
// in build.gradle
dependencies { ...
            implementation 'com.squareup.picasso:picasso:2.71828’
}

<!-- in AndroidManifest.xml -->
 <uses-permission android:name="android.permission.INTERNET" />
```

Displaying a web photo

- In your app's Java code, write:

```
Picasso.get()
        .load("url")
        .into(ImageView);
```

- Example:

```
 // show a photo
ImageView img = (ImageView) findViewById(R.id.photo);
Picasso. get()
.load("https://upload.wikimedia.org/wikipedia/commons/d/d4/Siberian_Tiger_by_Malene_Th.jpg")
.into(img);
```

Picasso image methods

Method	**Description**
centerCrop()	center and crop image inside view
centerInside()	resize image proportionally inside view
error(*id*)	show given drawable as error
fetch()	download image in the background
fit()	resize image to fit view bounds
get()	return image as a Bitmap
into(*view*)	puts image into given view
placeholder(*id*)	show given drawable while loading
resize(*width*, *height*)	change image size in pixels
rotate(*degrees*)	rotate clockwise
tag("*tag*")	attaches a "tag" to a loading image (useful for bulk operations shown later)
transform(*trans*)	apply complex transformations

Picasso image methods

Method	Description
cancelRequest(*view*)	abort any image loading in that view
cancelTag("*tag*")	cancel all images with given tag

invalidate("*url*")
invalidate(*File*)
	flush out cache of given image,
	so it will be re-downloaded the next time

load("*url*")
load(*id*)
load(*File*)
	load an image from various sources

pauseTag("*tag*")	pause all image loads for given tag
resumeTag("*tag*")	unpause all image loads for given tag
shutdown()	stop entire Picasso system
with(*context*)	use given activity/fragment as context

Ion library

- **Ion** is a library to make it easier to download files from the web.
  - https://github.com/koush/ion

- To add Ion to your project:

in build.gradle

```
dependencies {
        ...
        implementation 'com.koushikdutta.ion:ion:2.+’
}
```

```
<!-- in AndroidManifest.xml -->
<uses-permission android:name="android.permission.INTERNET" />
```

Downloading a web file

- In your activity code, write:

```
Ion.with(this)
        .load("url")
        .asType()
        .setCallback(new FutureCallback<Type>() {
        public void onCompleted(Exception e, Type result) {
                // code to process the result
                }
        });
```

Ion download example

- // grab a text file and log its contents

```
Ion.with(this)
        .load("http://www.example.com/notes.txt")
        .asString()
        .setCallback(new FutureCallback<String>() {
                public void onCompleted(Exception e, String result) {
                Log.v("ion", result);
                }
        });
```

- other types: asJsonObject, asByteArray

Ion to post data to a web server

- // grab an image file

```
Ion.with(this)
    .load("https://example.com/submit")
    .setBodyParameter("username", "jsmith12")
    .setBodyParameter("password", "123456")
    .asString()
    .setCallback(new FutureCallback<String>() {
    public void onCompleted(Exception e, String result) {
        Log.v("ion", result);
        }
    });
```

- can be used to submit form data to web servers / REST APIs

Other Libraries

- **Android-Bootstrap** is a library that provides some good-looking customizable widgets not normally available in Android.
  - https://github.com/Bearded-Hen/Android-Bootstrap

- **ButterKnife** is a popular library intended to simplify usage of Android widgets and events in Java code.
  - written by Jake Wharton
  - http://jakewharton.github.io/butterknife/

- **SwipeStack** is a library that helps you make a stack of views that look like cards that you can "swipe" left or right.
  - https://github.com/flschweiger/SwipeStack

- An ambitious Android user named *daimajia* has created several libraries, including one to do **animation effects** on Views.
  - https://github.com/daimajia/AndroidViewAnimations