

# Android App Course v3

  

## Welcome

### ▼ Labs

Lab 1: Hello World

Lab 2: Joke List

Lab 3: Joke List 2.0

Lab 4: Joke List 3.0

**Lab 5: Joke List 4.0**

Lab 6: WalkAbout

Lab 7: App Rater

## Resources

## How-to's

## Sitemap



[Java Essentials for  
Android Course](#)

[Labs](#) >

## Lab 5: Joke List 4.0

# Intro

This lab will be a continuation of Lab 3. You will be given a solution to Lab 3 so you may begin Lab 5 even if you did not complete all of Lab 3. You will be uploading individual jokes to a pre-existing web server, and downloading all the jokes on the web server. It is important to note that this lab is meant to be done in order, from start to finish. Each activity builds on the previous one, so skipping over earlier activities in the lab may cause you to miss an important lesson that you should be using in later activities.

# Objectives

**At the end of this lab you will be expected to know:**

- *How to establish Http connections.*
- *How to use AsyncTask.*

# Activities

For this lab we will be extending the "Joke List" application that you worked on in Lab 3. The previous app you created functioned completely on its own, but when the orientation or state of the device was changed the jokes disappeared, replaced by the original unrated prepopulated jokes. This version of the app will add a persistence layer to the previous version to fix these issues and more. It will allow Jokes that are added to be saved to a database as well as maintain application state throughout the life cycle of the application. All tasks for this lab will be based off of this application. Over the course of the lab you will be iteratively refining and adding functionality to the Joke List app. With each iteration you will be improving upon the previous iteration's functionality.

### **IMPORTANT:**

You will be given a Skeleton Project to work with. This project contains all of the java and resource files you will need to complete the lab. Some method stubs, member variables, and resource values and ids have been added as well. It is important that you not change the names of these methods, variables, and resource values and ids. These are given to you because there are unit tests included in this project that depend on these items being declared exactly as they are. These units test will be used to evaluate the correctness of your lab. You have complete access to these test cases during development, which gives you the ability to run these tests yourself. In fact, you are encouraged to run these tests to ensure that your application is functioning properly.

## **1. Setting Up**

### **1.1 Setting Up the Project**

To begin, download and extract the Lab 5 skeleton project for the

JokeList application from Polylearn. This is virtually the same project stub as you used for Lab 4, but it has some pointers on what code you need to write.

- Extract the project, making sure to preserve the folder structure.
  - *Take note of the path to the root folder of the skeleton project. You may prefer to extract it to your Eclipse workspace directory.*
- **Note: If you have a fully functional lab 3 implementation, you can use that as a basis and add new methods, classes, features and resources found in the skeleton project to it, or vice-versa.** Best to use a code base you're familiar with, provided it works as intended.

Next you will need to set up the Joke List Android project for this app. Since the skeleton project was created in Eclipse, the easiest thing is to import this project into Eclipse.

- Select **File -> Import....**
- In the Import Wizard, expand **General** and select **Existing Projects into Workspace**. Click **Next**.
- In the Import Project wizard, click **Select root directory** and click Browse. Select the root directory of the skeleton project that you extracted. Click Open and then Finish.
- Click on the project name in the Package Explorer. Select File -> Rename and change the name of your project to **lab5<userid>** where <userid> is your user id (e.g. jsmith).
- Check to make sure the package names

are **edu.calpoly.android.lab5** (**src** folder)  
and **edu.calpoly.android.lab5.tests** (**test** folder).

Make sure that your project is targeting the latest version of Android but supporting API 10 and higher.

- In the Manifest file, find the **uses-sdk** XML component and change **android:targetSdkVersion** to the latest API version (17 at the time of writing this lab).  
If **android:minSdkVersion** is not set to 10, do so.

Change the author name in **strings.xml** to **<userid>**. Non-Cal Poly students can change this to an appropriate username handle of their choice.

If you haven't set up your project to use [ActionBarSherlock](#) (ABS), do so. If you already have the library project for ABS, you should be able to skip this step.

- Use [the setup step from Lab 3](#) as a guide for downloading and creating the ABS library project and adding ABS as a library project to your imported stub project under **Properties** - **> Android** - **> Library**.

\*\*\*The Unit tests have not been updated for this lab. You may or may not find them useful.

Run the following three Unit Test classes (located under the **test** folder) as Android JUnit Tests:

- **AdvancedJokeListTest2**
- **AdvancedJokeListAddFilterTest**
- **AdvancedJokeListAddFilterDeleteTest**

These Unit Tests are the same tests from the downloadable Lab 3 stub project. They should all pass. If not, double-check your code (if it's your old code from lab 3) and make sure it works as intended according to [Lab 3](#).

## 1.2 Familiarize Yourself with the Source Code

The skeleton project that has been given to you for this lab contains a fully functional solution to Lab3. In particular, the AdvancedJokeList, JokeListAdapter, Joke, and JokeView classes are fully functioning classes from Lab 3. Additionally, many completed XML resource files (drawable, layout, menu) have been supplied as well. Most of the code in the new

It is a good idea, and good practice, to read through the source code. See how it compares to your implementation of Lab 3. It is especially important to do this if there were any parts of Lab 3 that you were not able to complete. The rest of the Lab will require you to update this source code to make use of Data Persistence so it is critical that you are familiar with it.

- Make sure the application runs and functions properly according to [Lab 3](#). If you did not complete Lab 3, make sure you understand how the code base works, what State Lists are, how the Action Bar works, etc.

## 2. Establishing Http Connections

You will establish Http connections to a web server that is provided to you. You are to add two features to your app:

- the ability to upload a single joke to the server
- the ability to download all the jokes from the server



You may use the instructions in **section 5** of <https://sites.google.com/site/androidappcourse/labs/lab-3> to help you, but note the following differences:

"Upload" will be a menu item in the Context Menu.

"Download" will be a menu item on the Action Bar.

Network connections must be completed off the UI thread. Use [AsyncTask](#) for this.

`uploadJokeToServer()` has already been implemented for you as an example. You just need to call `uploadJokeToServer()` from the appropriate Context Menu.

Hint 1: When downloading jokes, return an `ArrayList` of jokes from the server. Pass this `ArrayList` to `onPostExecute` where you can loop through all the jokes and call `addJoke()` on each one.

Hint 2: If you make your `AsyncTask` an inner class, you can use `AdvancedJokeList.this` instead of `this` to reference the context of the outer

class. (e.g. **Toast.makeText(AdvancedJokeList.this, toastText, Toast.LENGTH\_SHORT);**)

### 3. Deliverables

To complete this lab you will be required to:

1. Put your entire project directory into a .zip or .tar file, similar to the stub you were given. Submit the archive to PolyLearn. This effectively provides time-stamped evidence that you submitted the lab on time should there be any discrepancy later on in the quarter. The name of your archive should be **lab5<userid>** with a **.zip** or **.tar** extension. So if your username is **jsmith** and you created a zip file, then your file would be named **lab5jsmith.zip**.
2. Load your app on a mobile device and bring it to class on the due date to demo for full credit.
3. Complete the survey for Lab 5: <https://www.surveymonkey.com/s/436F13Lab5>

Primary Authors: James Reed

Adviser: [Dr. David Janzen](#)

### Comments

Commenting disabled due to a network error. Please reload the page.

You do not have permission to add comments.

