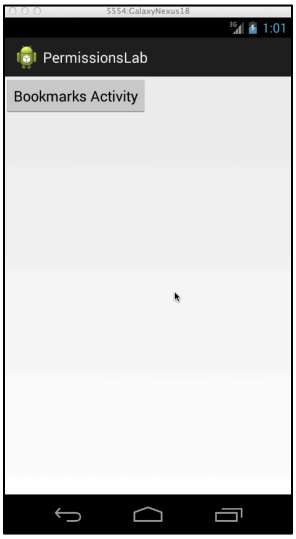
Lab – Permissions

Objectives:

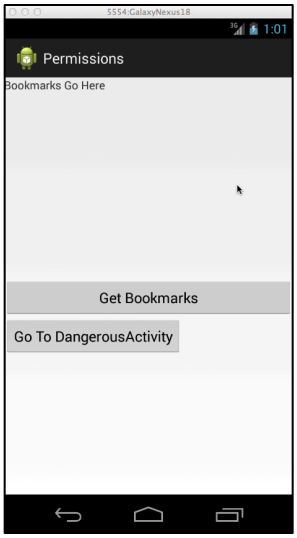
Familiarize yourself with Android Permissions. Create applications that use, define and enforce Android Permissions.

Exercise A: Using Permissions

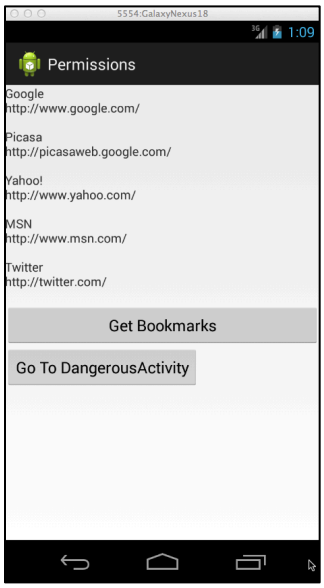
This exercise uses Permissions so that it can load protected content. The application is called PermissionsLab and its main Activity is called ActivityLoaderActivity. This Activity's user interface displays a Button labeled "Bookmarks Activity."



When the user clicks this Button, the application will start a new Activity called "BookmarksActivity." That Activity's user interface is shown below.



This activity presents a TextView that initially displays the words, "Bookmarks Go Here." It also presents a Button labeled, "Get Bookmarks," and another Button labeled, "Go To DangerousActivity." When the user presses the "Get Bookmarks" Button, the application retrieves the user's Browser bookmarks and then displays them in the TextView, as shown below.



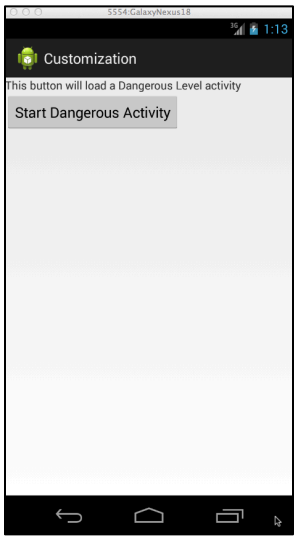
Android stores Browser bookmarks in a ContentProvider. We haven't discussed ContentProviders in detail yet, but the application skeleton includes all the code needed to query this ContentProvider. In order for this code to work however, your application must have permission to read the Browser Bookmarks. In order to complete this assignment you'll need to find the specific permission you need. See the documentation for more information.

Exercise B: Defining and Enforcing Custom Permissions

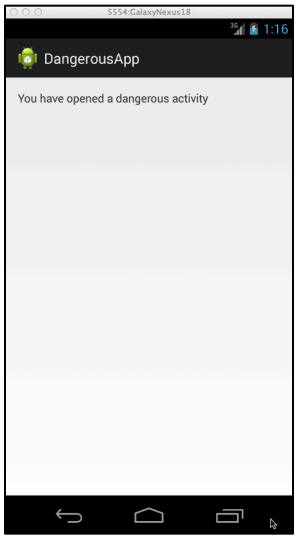
In this exercise you'll define, enforce and use permissions so that your application can access a separate, permission-protected application, called DangerousApp. You will build your solution to this exercise by extending your solution to Exercise A.

When the user clicks on the Button (shown above) labeled “Go To DangerousActivity”, an Activity called “GoToDangerousActivity” will be started. That Activity’s user interface appears below.

When the user clicks on the “Start Dangerous Activity” Button, this Activity will use an



Implicit Intent with the **action**, "course.labs.permissions.DANGEROUS_ACTIVITY", to start the “DangerousApp.” As shown below, that app will simply display a TextView, containing the words, “You have opened a dangerous activity.”



To implement the DangerousApp application, you will need to import and modify a separate Android application project that contains a single Activity called DangerousActivity. The application will define and enforce its own custom permission, “course.labs.permissions.DANGEROUS_ACTIVITY_PERM”, which will have a “dangerous” protection level. See the documentation for more information. You will also specify an intent filter for the DangerousActivity of the DangerousApp that matches the Implicit Intent that the PermissionsLab you define to start the DangerousActivity.

Implementation Notes:

1. Download the assignment's download zip file. It contains two projects in the Skeletons directory. The PermissionsLab is in PermissionsLab.zip and the DangerousApp is in DangerousApp.zip. There is also a TestCases directory that contains, the PermissionsLabTest project in PermissionsLabTest.zip. Import these projects into your IDE. In Eclipse, you can do this by selecting File>Import>General>Existing Projects Into Workspace. Then use the Browse button and navigate to a specific .zip file containing a particular project.

2. For Exercise A:
    a. In the PermissionsLab's ActivityLoaderActivity.java, find the comment containing the String TODO in the startBookMarksActivity() method. Start the BookmarksActivity.

    b. In the PermissionsLab's BookmarksActivity.java, find the comment containing the String TODO in the startGoToDangerousActivity() method. Start the GoToDangerousActivity.

    c. In the PermissionsLab's AndroidManifest.xml, find the comments containing a TODO String. Where indicated, add the appropriate uses-permission element so that this application can read the Browser bookmarks.

3. For Exercise B:
    a. In the DangerousApp’s AndroidManifest.xml, find the comments containing a TODO String. Where indicated, define and enforce a new permission named, “course.labs.permissions.DANGEROUS_ACTIVITY_PERM”, that has a dangerous protection level.
    b. In the DangerousApp’s AndroidManifest.xml, find the comments containing a TODO String. Where indicated, add Intent Filter information so that the DangerousActivity of this application can be started by an

implicit Intent, having the Action, "course.labs.permissions.DANGEROUS_ACTIVITY"

c. In the AndroidManifest.xml file for the PermissionsLab, find the comments containing a TODO String. Where indicated, add the appropriate uses-permission element so that this application can start the DangerousApp.

Testing and Submission:

The test cases for this Lab are in the PermissionsLabTest project. You can run the test cases either all at once, by right clicking the project folder and then selecting Run As>Android Junit Test, or one at a time, by right clicking on an individual test case class (e.g., TestBookmarks.java) and then continuing as before. The test classes are Robotium test cases. You will eventually have to run each test case, one at a time, capture log output, and submit the output to Coursera. These test cases are designed so that if you pass the test case, your submission should pass as well. The TestBookmarks test case should output exactly 3 Log messages. The TestDangerousApp test case should output exactly 3 Log messages.

As you implement various steps of the Lab, run the test cases every so often to see if you are making progress toward completion of the Lab.

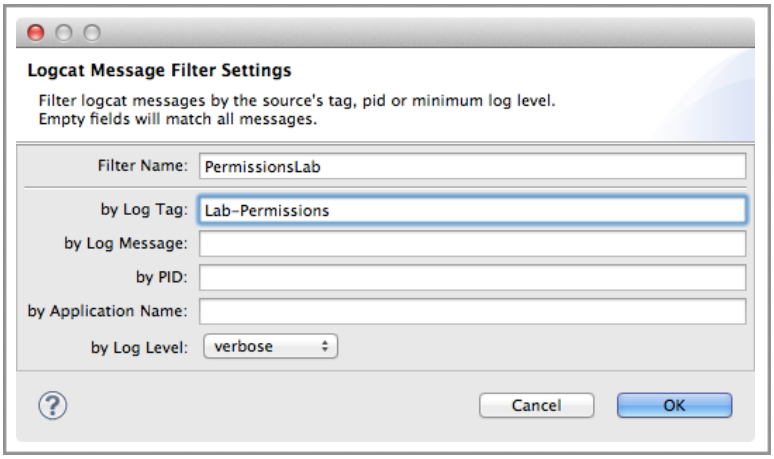
Warnings:

1. These test cases have been run on the emulator using a Galaxy Nexus AVD with API level 18. To limit configuration problems, you should test your app against a similar AVD.
2. The TestBookmarks test case assumes that there is at least one Browser Bookmark, with "http" in its URL.
3. The TestDangerousApp test case requires that you've installed both the PermissionsLab and the DangerousApp applications. Remember that each time you modify the DangerousApp you need to reinstall it.
4. The TestDangerousApp test case causes the DangerousApp to start. Due to Android security policies, Robotium test cases cannot test multiple applications. So this test starts up the DangerousApp, but can't interact with the PermissionsLab application after that. What this means for you, is that after running TestDangerousApp you must manually exit the DangerousApp (for example, by hitting the back button). If you don't and then try to run another test case, Eclipse gets completely stuck.

Once you've passed all the test cases, follow the instructions on the Assignment page to submit log information to the Coursera system for grading.

Tips: Saving a LogCat filter.

1. In the LogCat View, press the green "+" sign to "add a new LogCat filter."
2. A dialog box will pop up. Type in the information shown in the screenshot below.
3. A saved filter called, "PermissionsLab" will appear in the LogCat View.



Tips: Running your test cases and capturing your LogCat output for submission.

1.For each test case, clear the LogCat console by clicking on the "Clear Log" Button (the Button with the red x in the upper right of the LogCat View).
2.Then right click on an individual test case file in the Project Explorer. Run As -> Android JUnit Test.
3.When the test case finishes, if it's not already selected, click on the "PermissionsLab" filter in the left hand pane of the LogCat View.
4.Select everything within the LogCat View (Ctrl-A or Cmd-A) and press the "Export Selected Items To Text File" button (floppy disk icon) at the top right of the LogCat View.
5.Submit this file to the Coursera system.