

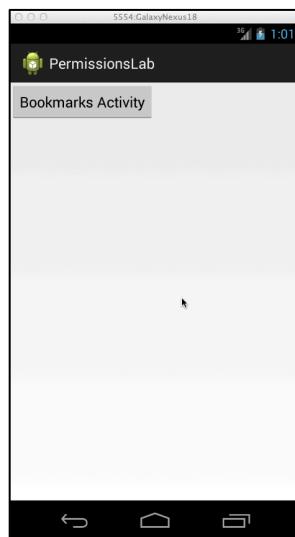
Lab – Permissions

Objectives:

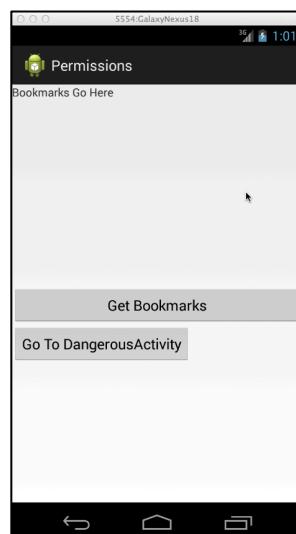
Familiarize yourself with Android Permissions. Create applications that use, define and enforce Android Permissions.

Exercise A: Using Permissions

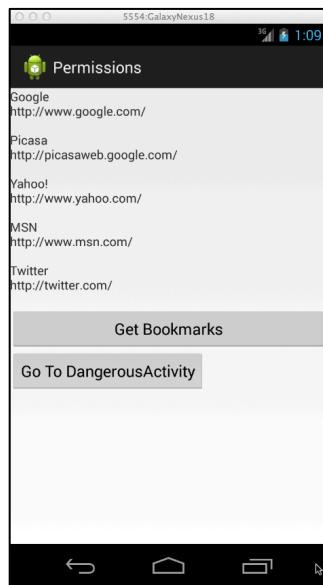
This exercise uses Permissions so that it can load protected content. The application is called Lab3b_PermissionsLab and its main Activity is called ActivityLoaderActivity. This Activity's user interface displays a Button labeled “Bookmarks Activity.”



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.

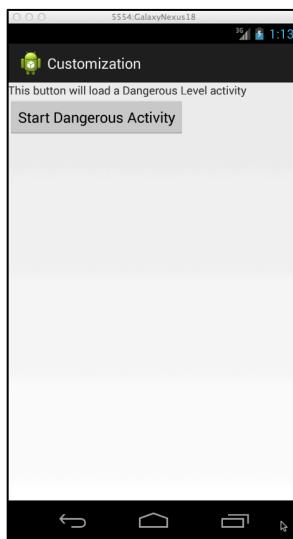


Android stores Browser bookmarks in a ContentProvider. We haven't discussed ContentProviders in detail yet, but the application skeleton includes all the code needed to query this ContentProvider. In order for this code to work however, your application must have permission to read the Browser Bookmarks.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

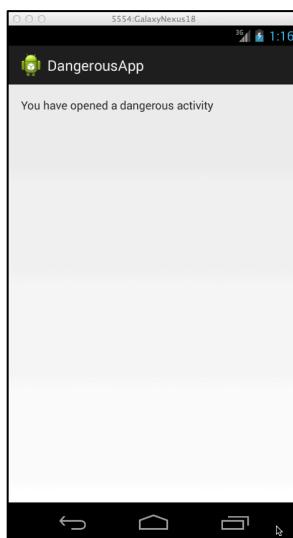
Exercise B: Defining and Enforcing Custom Permissions

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 Lab3b_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 <http://developer.android.com/guide/topics/manifest/permission-element.html> for more information. You will also specify an intent filter for the DangerousActivity of the Lab3b_DangerousApp that matches the Implicit Intent that the Lab3b_PermissionsLab you define to start the DangerousActivity.

See the screencast in the download package to see the app in action.

Implementation Notes:

1. Download the assignment's download zip file. It contains two projects: the Lab3b_PermissionsLab and the Lab3b_DangerousApp. The test cases are in the Lab3b_PermissionsLabTest project. 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 Lab3b_PermissionsLab's ActivityLoaderActivity.java, find the comment containing the String TODO in the startBookMarksActivity() method. Start the BookmarksActivity.
 - b. In the Lab3b_PermissionsLab's BookmarksActivity.java, find the comment containing the String TODO in the startGoToDangerousActivity() method. Start the GoToDangerousActivity.
 - c. In the Lab3b_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 Lab3b_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 Lab3b_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 Lab3b_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 Lab3b_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 and then continuing as before.

To Submit Files:

1. Your project files must be compressed in a zip file named ActivityLabSubmit.zip.
2. The zip file should have the following directory structure:
PermissionsLabSubmit/
 -Dangerous/
 - AndroidManifest.xml
 -Permissions/
 - ActivityLoaderActivity.java
 - BookmarksActivity.java
 - GoToDangerousActivity.java
 - AndroidManifest.xml
3. Each new submission will overwrite any previous submissions.

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 Lab3b_PermissionsLab and the Lab3b_DangerousApp applications. Remember that each time you modify the Lab3b_DangerousApp you need to reinstall it.
4. The TestDangerousApp test case causes the Lab3b_DangerousApp to start. Due to Android security policies, Robotium test cases cannot test multiple applications. So this test starts up the Lab3b_DangerousApp, but can't interact with the Lab3b_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 have passed all the test cases, follow the instructions on the Assignment page to submit your work.