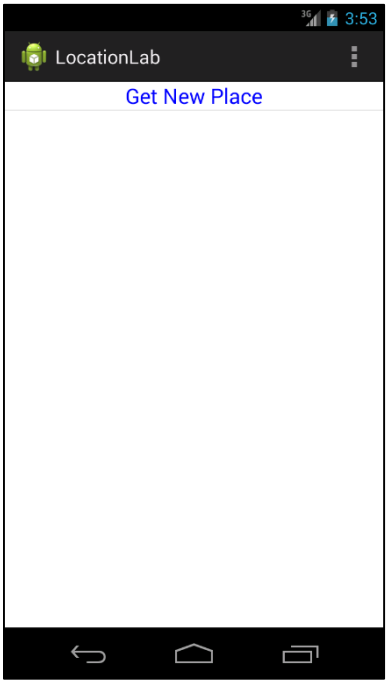
# Content Provider Lab

*Use Content Providers to allow your application to save, access and share data.*

## Objectives:

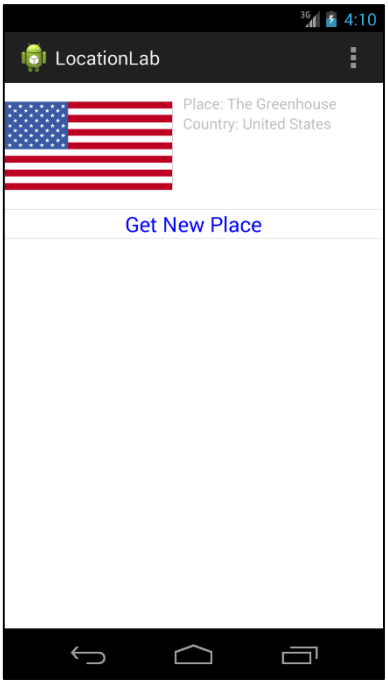
In this week's lab, you'll learn more about Content Providers. Content Providers are used to store, manage and share data across applications. Often, but not always, this data is stored in an SQLite database.

This week's lab will build off last week's Location lab. This application will have the same UI elements and behaviors as last week's Lab. This application, however, manages a user's Place Badges using a ContentProvider. It will therefore need to access the Content Provider when it adds or displays Place Badges.




For instance, if the user clicks on the Footer and the application does not already have a Place Badge for a location within 1000 meters of the user's current location, then the application acquires the data needed to create the Place Badge. The PlaceViewAdapter will then insert this newly collected information into a Content Provider.

Once these processes are complete, the new Place Badge will appear in the ListView, exactly as it did before.




Because ContentProviders persist data across application sessions, when the application initially starts up, it should get all existing Place Badges from the ContentProvider and display them on the screen.

## Implementation Notes:

1. Download the application skeleton files and import them into your IDE. This Lab involves three projects
    a. ContentProviderLabContentProvider - The ContentProvider for managing Place Badges.
    b. ContentProviderLabUser - The user-facing app code that collects Place Badges and stores them in the ContentProvider.
    c. ContentProviderLabTest - The test cases for this Lab.
2. **IMPORTANT NOTE: T**here are some small changes from last week's project files. As a result, we strongly encourage you to copy your code changes from last week's Lab into the corresponding locations in this week's skeleton files, rather than trying to simply add new changes to last week's Lab files.
3. Complete the TODO items, which are mainly in the PlaceViewActivity.java and PlaceViewAdapter.java files. Your geonames.org username may also need to be updated in PlaceDownloaderTask.java.

## Testing:

The test cases for this Lab are in the ContentProviderLabTest 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., TestOneValidLocation.java) and then

continuing as before.

**Warnings:**

1. These test cases have been tested on a Galaxy Nexus AVD emulator with API level 18 and with 100Mb of external storage. To limit configuration problems, you should test you app against a similar AVD. In particular, you must have external storage available on your device.
2. Our MockLocationProvider relies on a method that was included in API level 17. Therefore, the TestCases will fail to compile on earlier platforms.
3. You will need to go into your device's Developer Options and make sure that you've enabled "Allow Mock Locations."
4. During testing you should not provide your own location information. To inject locations you will need to click on menu options, specifically, "Place One," "Place Two" and "Place No Country."
5. If you want to download data from the Internet, you will need to create an account at http://www.geonames.org/login. Your username will need to be updated in PlaceDownloaderTask.java.
6. Make sure that you install the ContentProviderLabContentProvider before you install the ContentProviderLabUser. The projects will not run properly otherwise.

## Submission

To submit your work you will need to submit the ContentProviderLab project files we've asked you to modify. These files should be stored in specific directories as described below and then compressed in a zip file. Then you will submit this zip file to the Coursera system. The automatic grading system will test your submission and give you feedback. This process may take some time, especially if many students are submitting at the same time.

To make sure your submission is correctly graded, pay attention to the following aspects:

1. Your project files must be compressed in a zip file named ContentProviderLabSubmit.zip.
2. When decompressed, your submission should contain one top-level directory named ContentProviderSubmit. Inside that directory there should be one directory named ContentProviderLab containing the following files: PlaceViewActivity.java, PlaceViewAdapter.java and PlaceDownloaderTask.java.

The directory structure of the unzipped submission file should be as follows:

```
ContentProviderLabSubmit /
        ContentProviderLab/
                PlaceViewActivity.java
                PlaceViewAdapter.java
                PlaceDownloaderTask.java
```