ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 5

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your Android project as follows:
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. When you submit the assignment, compress your exported Android project into a single zip file. The format of compressed file name is InClassAssignment#.zip
   b. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 5 (100 Points)

In this assignment you will get familiar with Android Concurrency, HTTP connections and XML parsing. You will build a simple weather application.

Important App Requirements:

1. Create a new android project called “In Class 5”.
2. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 17**. The app should display correctly on 3.2” QVGA (ADP2) (320x480: mdpi). Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.
3. You will be using layout files, and strings.xml to create the required user interfaces. The layout XML file can be modified through the raw xml, or through the GUI tools provided within eclipse.
4. All API calls, image downloading and decoding should be performed using a worker thread or (or AsyncTask) and your code should not block the main thread.
5. Your code should use standard naming conventions, such as, uppercase class names, and lower case variable/method names. Also your variable and method names should be descriptive of the data or action performed.

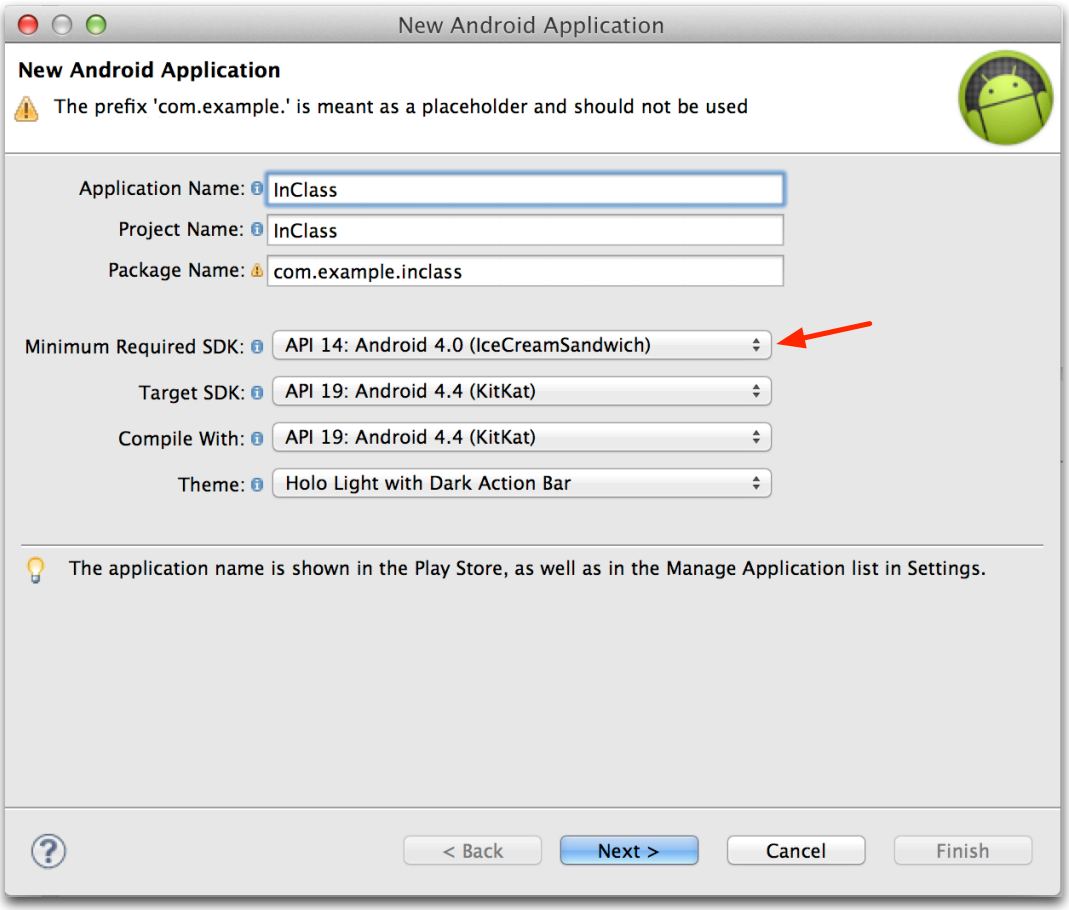


Fig 1. Choosing Minimum Required SDK to 14

Initial Setup and API Description

You should use the OpenWeatherMap api (http://openweathermap.org/current) for getting the weather information. The api of interest is the Current Weather Data api which is based on the city name and country name. For information related to the api please check http://openweathermap.org/current. The API details is as follows:

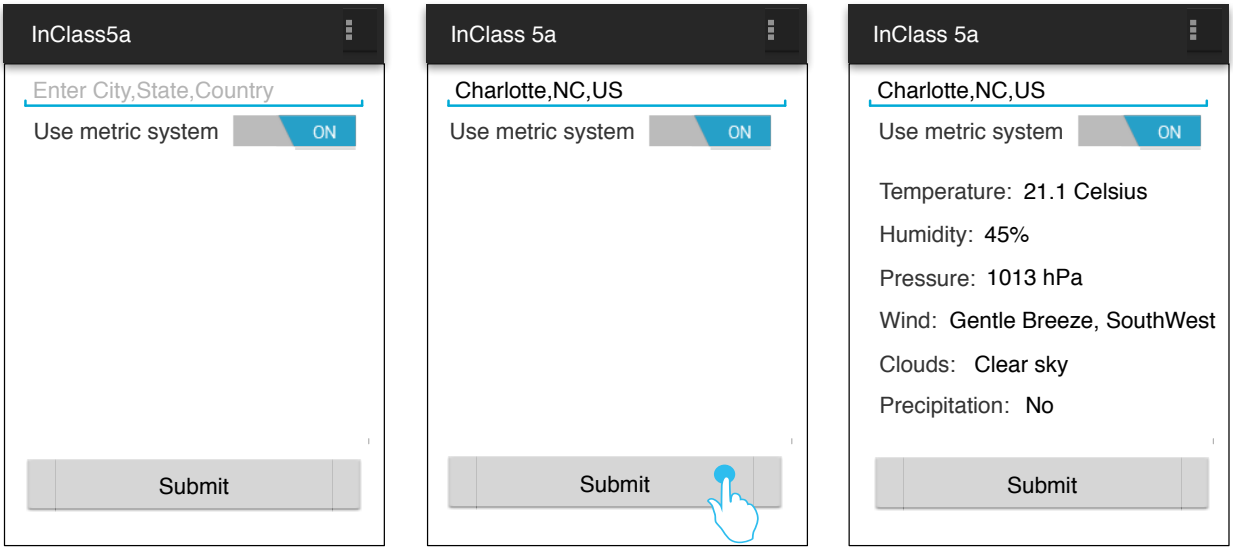
- Endpoint: http://api.openweathermap.org/data/2.5/find
- Arguments (GET Method)
  - q: this is the city name and country name for example, `Charlotte,US`.
  - `units`: can be set to either `metric` or `imperial`.
  - `mode`: should be set to `xml`.

For example to retrieve the weather for Charlotte, in imperial format the url should be setup as follows:
http://api.openweathermap.org/data/2.5/find?q=Charlotte,US&units=imperial&mode=xml

The response will be as follows :

```
<cities>
    <calctime>0.1052</calctime>
    <count>1</count>
    <mode>name</mode>
    <list>
        <item>
            <city id="4460243" name="Charlotte">
                <coord lon="-80.843132" lat="35.227089"/>
                <country>US</country>
                <sun rise="2014-09-22T11:11:49" set="2014-09-22T23:19:52"/>
            </city>
            <temperature value="69.12" min="66.2" max="71.6" unit="fahrenheit"/>
            <humidity value="46" unit="%"/>
            <pressure value="1019" unit="hPa"/>
            <wind>
                <speed value="5.62" name="Moderate breeze"/>
                <direction value="340" code="NNW" name="North-northwest"/>
            </wind>
            <clouds value="1" name="clear sky"/>
            <precipitation mode="no"/>
            <weather number="800" value="Sky is Clear" icon="01d"/>
            <lastupdate value="2014-09-22T15:17:07" unix="1411399027"/>
        </item>
    </list>
</cities>
```

Figure 2, XML Response



(a) Main Wireframe (b) Enter City,Country (c) Result displayed

Figure 3, Main Activity Wireframe

Part 1: Parsing XML Document using XML SAX Parser (80 Points)

The Activity UI should match the UI presented in Figure 2. Below are the requirements:

1. Create a new android project called “In Class 5a”.
2. The EditText will hold the query consisting of comma-separated values. See Figures 3(a) and 3(b).
3. You should use a separate thread to perform data retrieval from the server and data parsing. Do not use the Main Thread to perform these tasks. Use an AsyncTask or a Thread/Handler.
4. The “Use metric system” switch decides the value of the units parameter, if ON then units should be set to metric, else it should be set to imperial.
5. When the “Submit” button is tapped, the corresponding query request should be sent to the server to retrieve the xml document stream. If multiple cities are returned in the XML, only retrieve the first city.
6. Create a weather class containing string variables temperature, humidity, pressure, windSpeed, windDirection, clouds, and precipitation.
7. Implement an XML SAX Parser and pass the document stream to the parser. Parse the weather information and store it in a weather object.
8. Display the weather information in TextViews as shown in Figure 3(c).

Part 2: Parsing XML Document using XML Pull Parser (20 Points)

This part is similar to the part 1, below are the requirements:

1. Create a new android project called “In Class 5b”.
2. Implement the app in Part 1 but instead use the XML Pull Parser to parse and display the city’s weather information.