

GOOGLE MAPS AND RSS INTEGRATION IN ANDROID

Sabyasachi Patra¹, Karishma Velisetty² and Prathamesh Patel³

^{1,2,3} B.Tech Students, Department of Computer Science, Mukesh Patel School of Technology Management and Engineering, NMIMS University, Mumbai

sabya9205@gmail.com, karish159@gmail.com, pratham2392@gmail.com

Abstract

This paper describes the process of integrating Google Maps API V2 as well as RSS feeds into an Android project. Authors have made an application. The basic purpose of this application is to use it as an introductory application to get our hands dirty on native Java coding and embark upon our research domains; un-burdened and with a smoother paradigm of thoughts. It is also useful for any Android developer to get a command on the basic algorithms which deal with Google Maps and other very important client - based and efficiency - enabling offline applications. Basically, it's the textual modelling of our first Android based basic – level application.

Keywords - Android, RSS, Google Maps, API Key, TOI, Map Fragments, Android SDK, SAX Parser

1. INTRODUCTION

Google Maps for mobile runs on smartphones with over 54% of users using it. It allows users explore the world with rich maps provided by Google. Identification of locations with custom markers, augmenting the map data with image overlays, embedding one or more maps as fragments, and many such implementations help the average Android user. Using the core engine and the map/satellite images hosted by Google, such tools can introduce custom location icons, location coordinates and metadata, and even custom map image sources into the Google Maps interface.

Steps to integrate the Google Maps V2:

1. Install Google Play service **Error! Hyperlink reference not valid.**
2. Get the *Google Maps API Key* **Error! Hyperlink reference not valid.**
 - Create an API Project **Error! Hyperlink reference not valid.**
 - Obtain an API Key **Error! Hyperlink reference not valid.**
3. Add API key to AndroidManifest file
 - Update the **AndroidManifest** file with other settings
 - Display a **MapFragment**

Steps followed after integration:

1. Start the Android SDK Manager
2. Choose to install Google Play services from the 'extras' category:
After the sdk manager completes the installation, go to
`<android_sdk_folder>/extras/google/google_play_services/libproject`
3. Copy google-play-services_lib to the location where you maintain your Android projects.

a. Getting the Google Maps API Key

Steps:

Generate an API Project:

- Navigate to Google APIs Console and create a new project
- Click service, obtain an API Key then Click on Create New Android Key
- Add the API key to Android Manifest file:

Open the Android Manifest file and add the following element as a child of application tag and replace the **YOUR_API_KEY with your real API key.**

```
<meta-data android:name="com.google.android.maps.v2.API_KEY"  
android:value="YOUR_API_KEY"/>
```

b. Updating the Android Manifest file with other settings

In order to use Google Maps Android API we need to declare a few permissions and specify that the application requires *OpenGL ES version 2*. This is basically an important requirement for the *Android Emulator*, on which the map would be displayed, when the application runs.

c. Displaying a Map Fragment:


```
List<RssFeedStructure>
rssStr ;
private RssReaderListAdapter _adapter;
SAXParserFactory spf =
SAXParserFactory.newInstance(); SAXParser sp =
spf.newSAXParser();
XMLReader xr =
sp.getXMLReader(); url = new
URL(feedUrl);
xr.setContentHandler(this);
xr.parse(new InputSource(url.openStream()));
```

3. ADDING ADVERTISEMENT TO AN ANDROID APPLICATION

1. Create an 'AdMob' account by browsing to <http://www.admob.co>**Error! Hyperlink reference not valid.**
2. Login to your account and follow the following steps to get an advertisement imported.
3. Click on sites & Apps tab on the top menu. Then click on Add Site/App button to add the application you want your advertisements to show up on
4. Select the Android platform.
5. **AdMob** will add your application to the list and ask you to download the **AdMob Android SDK**.
6. Now start your Eclipse IDE to import the downloaded SDK. Right click and choose properties.
7. Select "Java Build Path" option from the navigation pane to your left and then click on "libraries tab" to add the SDK file. Click on "Add external JARS..." button and locate the unzipped 'AdMob SDK JAR' file on your disk, and add it to your project.
8. If you successfully completed the previous step, you should now see the JAR file (along with other required ones) located under, **Referenced Libraries folder** inside your project.
9. Once the SDK is added to your project, open the *AndroidManifest.xml* file located in your project. In order to use AdMob in your application you must include the following xml Just before the </application> tag enter a new activity for **com.google.ads**. AdActivity by adding this xml :

```
<activity android:name="com.google.ads.AdActivity"
android:configChanges="keyboard|keyboardHidden|orientation|screenLayout/uiMode|screenSize/smallestScreenSize" />
```

AdMob SDK work.
10. Finally, to add an 'AdMob' advertisement to your application, open your desired layout xml file. For this tutorial we will just add it to our main.xml located under res/layout folder of our project. Then add this following xml where you like to show your ad in your layout. In <LinearLayout> tag add

```
xmlns:ads=http://schemas.android.com/apk/lib/com.google.ads.Error! Hyperlink reference not valid.
<com.google.ads.AdView android:id="@+Error! Hyperlink reference not valid.
ads:loadAdOnCreate="true" > </com.google.ads.AdView>
```

4. CONCLUSION

The application made by our team is not only restricted to the implementation of features such as Google Maps and RSS integration. We have also included an inherent client – server model in the form of embedding Google forms in our application. The results of the data input from the client are stored in the Google drive server where an authorized person can view it and hence analyze it. Our paper focus due to academic constraints revolves around Map and RSS feeds implementation.

5. FUTURE SCOPE

The client server model which we have inherently implemented in our application is powered by the Google server, i.e. the inputs from the mobile device are directly fed to the Google server wherein the authorized user can view the response. This can be extended by making the **application online** w.r.t. a particular server machine (for E.g. PHP) and hence run the application

Dynamic user interaction and Map Markers can be implemented. Location updation can be done dynamically at the server side.

6. ACKNOWLEDGMENT

This research paper is made possible through the help and support from many people, and in essence, all sentient beings. Especially, please allow me to dedicate my acknowledgment of gratitude towards our college Librarian Mr. Pradip Das and his team. I would like to thank Mr. Anand Gawadekar of the NMIMS IEEE Committee due to which we could get all requested references seamlessly without any trouble. Finally, sincere

thanks to our college and Computer Science department for allowing us to enter the field of research in our so important final year of B.Tech. Special thanks to our H.O.D, Prof. Dharendra Mishra.

REFERENCES:

- 1) Chao Wang, Wei Duan, Jianzhang Ma, Chehui Wang, The Research of Android System Architecture and application programming, 2011, pp 6.
- 2) Divya Sambasivan, Nikita John, Shruthi Udayakumar, Rajat Gupta, GENERIC FRAMEWORK FOR MOBILE APPLICATION DEVELOPMENT, 2011.
- 3) Want.R., iPhone: Smarter than the Average Phone, Pervasive Computing, IEEE, Volume, 9, 19 July 2010, pp 6.
- 4) Butler, M., Android: Changing the Mobile Landscape, Pervasive Computing, IEEE, Volume 10, 23 December 2010, pp 4-7.
- 5) Kumar, S.Qadeer, M.A.Gupta, A. Dept. of Computer. Eng., Aligarh Muslim Univ., Aligarh, India, Location based services using android (LBSOID), Internet Multimedia Services Architecture and Applications (IMSAA), 2009 IEEE International Conference, 29 March 2010, pp 1.
- 6) Android SDK. <http://developer.android.com/sdk/index.html> [EB/OL], 2009.
- 7) Sayed Y.hashimi, Satya Komatineni. Pro Android[M]. Apress. 2009.
- 8) Mark L.Murphy, Beginning Android.[M]. Apress. 2009.
- 9) Android Developers, <http://www.androidin.com>

IMAGES INDEX



Figure 1: G-MAP as Map View



Figure 2: G-MAP as Satellite View



Figure 3: RSS Feeds from TOI Education (Tile View)

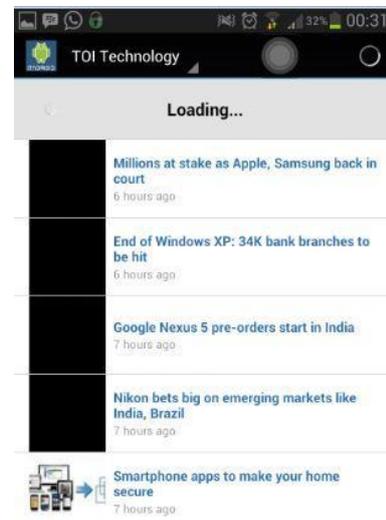


Figure 4: RSS Feeds from TOI Education (List View)

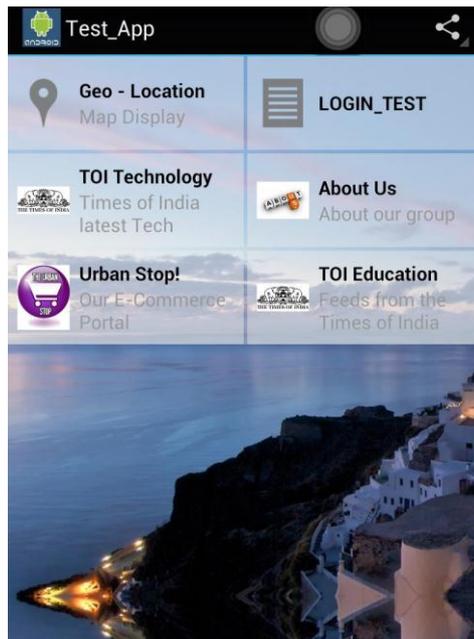


Figure 5: Application Homepage

AUTHORS BIOGRAPHY

Sabyasachi Patra is currently a student of NMIMS Mukesh Patel School of Technology Management and Engineering. His background throughout school as well as Junior College has been Science. He has a keen interest in Research, both in the domain of computers and philosophy; apart from which, Physics and Creative writing are also his primary pursuits of interest. His areas of interest in Computer Science broadly lie in the fields of Mobile Computing/implementations, Client – server web application development and Computer Architecture.

Karishma Velisetty is currently a student of NMIMS Mukesh Patel School of Technology Management and Engineering. Her background throughout school as well as Junior College has been Science. She has an interest for the process of Software development and Computer Science theories. Apart from Computers, she has interest in finance.

Prathamesh Patel is currently a student of NMIMS Mukesh Patel School of Technology Management and Engineering. His background throughout school as well as Junior College has been Science. His interests in the field of Computer Science lie in the field of application development and Websites.