Google Maps for Android Applications

SE3A04 – Tutorial

Andrew LeClair

Department of Computing and Software
Faculty of Engineering
McMaster University
Hamilton, Ontario, Canada
leclaial@mcmaster.ca

October 28/29, 2014

Outline

Helpful Resources

Android Developers
The Android SDK
http://developer.android.com/sdk/index.html

Google
Google Play Services SDK
http://developer.android.com/google/play-services/setup.html

Google
Google Maps API
https://developers.google.com/maps/documentation/android/

Getting Started

- To use the Google Maps API you must do three things:
  1. You will need to install all SDKs (Android and Google Play Services SDK)
  2. You **must** register your project with Google and get a certificate
  3. Add the required settings to your manifest
- After these three steps, you will be able to use the Google Maps API!

Installing the SDKs

- If you do not already have an IDE, download one
  - Eclipse is a good tool to begin with
- Download the Android SDK from the first website in *Helpful Resources*
- Download the Google Play Services SDK from the second website
- Install and **configure** the SDKs

Using an Emulator

- There exists resources that state it **is** possible
- It requires additional APKs
  - com.google.vending-20130716
  - com.google.android.gms-20130716
- The following website goes through steps for getting it to function
- `http://stackoverflow.com/questions/19372399/running-google-map-application-on-android-emulator`

Registering Your Project

- You will need your application's certificate (SHA-1 fingerprint) and package name
- You will most likely have a **debug certificate**
  - **Note** If you plan to release your application, your Android SDK will generate a new certificate for your "release" build. The Google services will need to be updated of this new certificate
- You can find your SHA-1 key with the keytool program located in your Java file under ..\jre7\bin

Registering Your Project

- Navigate to the Google API's Console
- Create a New Android Key using your SHA-1 fingerprint
- You will get a 40 character key
- Add this to your manifest – Then you're done!

Example

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

Map Objects

- Two ways to represent the map object: **MapFragment** and **MapView**
- Both are subclasses of their respective Android counterparts (MapFragment is a subclass of the Fragment class, etc.)
- Fragments are the more versatile of the two
- In one activity, there may exist many fragments
- The MapFragment class allows you to place a map in a Fragment

The MapFragment

- A fragment element can be added to the layout file by using the google play services
  - It may also be added through code; this is often more difficult

Example

```
<fragment
xmlns:android="http://schemas.android.com/apk/red/android"
android:name="com.google.android.gms.maps.MapFragment"
. . . />
```

Initialization

- The initial state that the map fragment starts in can be modified
- They can be modified in the XML attributes (or programmatically if that's how the fragment was created)
- Map type, initial camera starting location, initial zoom, etc.
- The namespace must be added to the XML document if you wish to change the initial state
  - If you wish to change the initial state programmatically, you must create a GoogleMapsOptions object

My Location

- The **My Location** layer provides the user with their currently location
- The layer itself does not provide any data
  - To get data it must be done programmatically through the Locations API
- Two app permissions must be changed to be able to request the data (coarse and fine locations)

My Location

- The function getLastLocation() can be used to get the current location

Example

```
Location mCurrentLocation;
mCurrentLocation = mLocationClient.getLastLocation();
. . .
```

Directions

- The **Google Directions API** calculates distances between locations
- Responds to **static** addresses, i.e. destinations known before-hand
  - If you wish a dynamic directions calculations, look at JavaScript API V3 Directions Services
- The API has the following limits in place:
  - 2500 requests per 24 hours
  - Up to 8 waypoints in each request
  - 2 requests per second
- **NOTE** These limits most likely will not affect you

Directions

- You'll be able to make a request in the form of:

Example

http://maps.google.com/maps/api/directions/output?parameters

- The output can either be json or xml
- For added security, you can use:

Example

https:// ...

Directions

- The origin and destination can be input as a string
- The directions services will geocode the string and convert it into **longitude** and **latitude** coordinates
- Optional parameters that can be selected can be found at:
  - `https://developers.google.com/maps/documentation/directions/`

Aside: Longitude and Latitude

- Co-ordinates on Earth are described by two values: their **longitude** and **latitude**
- The latitude is the horizontal lines of the globe, whereas the longtitude are the vertical
- They are quantified via degrees
- As an example, McMaster University is: 43.26°N, 79.92°W
- You can't input the cardinal direction, so represent N/E as positive, S/W as negative
- McMaster would then be 43.26°, -79.92°

Directions

Example of destination route

.../directions/json?origin=Toronto&destination=Montreal&key=API_KEY

- In this example, the following information is encapsulated in the http request:
  - The Origin is Toronto
  - The Destination is Montreal
  - The method of transport is driving (driving is default)
  - The response will be in JSON format
- The request can be modified by adding more fields, for example "&mode=bicycling"

JSON vs XML

- JSON is recommended because of the simplicity to parse

Example

"duration":
"value": 74384,
"text": "20 hours 40 mins" ,

- XML may be familiar, but it is more difficult to parse and possible to become "lost"

Example

<duration>
<value>74384</value>
<text>20 hours 40 mins</text>
</duration>

JSON Parsing

- The output is already in javascript so it is easy to parse in javascript

Example

```
for(i=0; i < myJSONResult.results.length; i++){
    myAddress[i] = myJSONResult.results[i].formatted_address;
}
```

- JSON contains multiple values, so it is wise to iterate over the length of the array
- If you have a particular value you wish to access, it is possible

XML Parsing

- Can use XPath to describe the nodes and elements within an XML document
  - The tags of the document form nodes, where the root node is the entire document
- XPath uses expressions to select elements within the document
- Java natively supports XPath
- The code is extremely laborous and convoluted
- If you have no experience with either, I suggest you begin with JSON because of its simplicity

Routes

- The Directions API returns the results within a JSON routes array
- Each element of the routes array contains a single result from the origin to destination
- The number of elements depends on how many legs there are
  - Legs themselves depend on if waypoints were specified

Routes

- Within a route element, you will find:
  - **Summary**: A textual description for the route
  - **Legs**: Array of information on the leg(s) (present for each waypoint, if no waypoints, there is only one leg)
  - **Waypoint_Order**: Indicates the order of the waypoints
  - **Overview_Polyline**: The polyline representation of the route
  - **Bounds**: The bounding box of the polyline
  - **Copyrights**: Copyrights text displayed for the route
  - **Warnings**: Warnings to be displayed when showing the directions

Legs

- The legs are a sub part of the route
- Each leg is determined by the waypoint, if no waypoints exist, the route consists of a single leg
- The leg contains the information that is important for metrics
  - Distance, Duration, Duration in traffic, arrival time, etc.

Summary

- Get the SDKs started, and your application registered so you can use the APIs!!
- A map can be contained within a view or fragment
- When mylocation is allowed, directions can be used
- The output from directions is JSON or XML, of your choosing
- The results can be parsed for the specific data you need

Questions

- Questions?