

DISTRIBUTED SYSTEMS  
AN AUTOMATED AND PERSONAL TOURISTIC TOUR  
APPLICATION USING GOOGLE MAPS API  
SECOND STEP : ANALYSIS OF EXISTING  
APPLICATIONS AND OF GOOGLE MAPS API

Quentin AUGRAIN  
Florent MALLARD

April 26, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Existing applications</b>	<b>3</b>
2.1	Bluebridge . . . . .	3
2.2	mTrip . . . . .	5
<b>3</b>	<b>Google Maps API</b>	<b>5</b>
3.1	Google Maps Android API . . . . .	6
3.2	Google Maps API Web Services . . . . .	7
3.3	Google Places API . . . . .	7

# 1 Introduction

We want to design an application which allows a tourist to optimize his visit of a city, offering him to select the monuments he wants to see and finding the best way to go to all of them. At first we also thought of a few additional features, such as displaying the distance between each monument in the list and the traveling time. In order to know what existing tourism applications already offer to do, and decide whether it's worth adding features to our application or not, we found some and analyzed them. In the following report we will firstly propose a description of the applications we found, their features, why we decided to implement them or not, and the technologies involved. Then we will also show the results of our analyze of Google Maps API, which we will use in our application. It will include what we can do with it, the ease of implementation and the services proposed by the API.

## 2 Existing applications

We managed to find some applications, but also platforms allowing you to build your own application.

### 2.1 Bluebridge

Bluebridge is a provider of tourism applications. It provides :

- Back-end technology (platform) to power your app
- Mobile App Studio CMS (product) to design your app
- Best-in-class app packages
- Professional services
- A message center to manage push notifications

The platform is a Software as a service, using its own technology, so we don't have any way to know how they are implemented. But the features proposed on the website are interesting, so we will developp them.

**GPS locating** As far as we thought, the GPS location is one of the main features when you want to implement an applications creating tours. Of course we can decide not to use it, and just display the best way between the monuments the user chose. This would mean that we assume that the user can find the first monument of the tour alone, and then will manage to find easily the way the application displays. The GPS location will allow the user to know where he is and

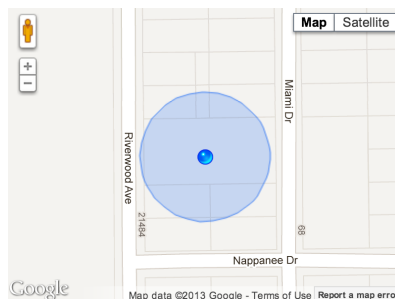


Figure 1: Google Maps displaying your location.

to confirm he goes in the right direction.

**GPS directions** This function is complementary with displaying the way between two monuments. Thanks to the GPS directions, the user can just be guided by the GPS, telling him which street he is currently in, where and when he should turn. Of course it will also announce his arrival to the monument he selected. It's not a compulsory feature, but it sure is a convenient one. Since

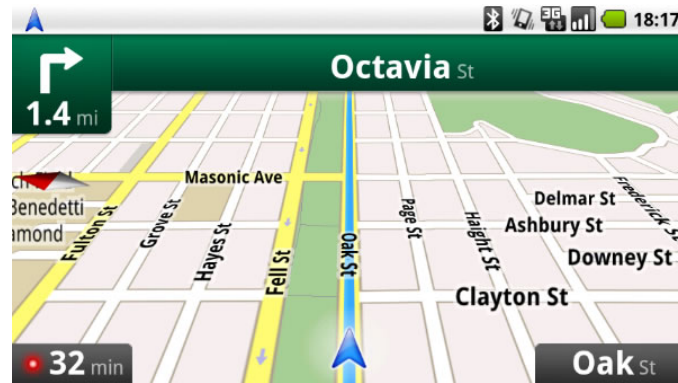


Figure 2: Google Maps displaying the navigation directions.

Bluebridge is not a real application, those are the main features we found. They are pretty classic so we won't present them in other applications. Let's head to the second application.

**Itinerary planner** It seems like the main idea of our application already exists. This feature is called Itinerary planner in Bluebridge, and give the users the option to set an attraction as “favorite”. When added as so, the attraction is added to a tab where users can plan their stay. It also gives the quicker route to the favorite attractions.

**Experience guides** This is a real little guide within the application. It can contain several things, such as :

- One-touch, turn-by-turn directions to attractions (interactive tours)
- Featured photos
- Content about local happenings
- Exclusive discounts and offers

In our application, only one seems to be interesting. We don't want our application to be a guide for visiting the place you are in, we just want the best way to join the places the user selected. But it can be nice, and also help the user to recognize the place, to display some pictures of the place the user is heading to. It would be a kind of preview that could better a little the design of the application, and for this reason it is worth thinking about implementing it.

Here were the most important and interesting features of the provider Bluebridge. We will now discover another provider which offer other types of features.

## 2.2 mTrip

mTrip is a company specialized in the development of tourism applications. Their applications have been downloaded over 2.5 million times, thanks to some special features like :

- Offline content
- Augmented reality
- Nearby places
- Currency converter
- Trip journal
- Sharing features

We won't be interested in all of them, only 3 of them will be developed in the following paragraphs.

**Offline content** We can note the offline content, that is really interesting because it allows the user to access data on the application without being connected to the Internet. Our application, and more generally tourism applications are meant to be used abroad, where roaming fees are high and you don't want to be connected too often. But the data has to be stored within the application, which can make it a big application to download.

**Augmented reality** Augmented reality will not be useful in our case, but the aim of this feature is to allow the user to visualize the place he wants to visit, or the way to go to it. Since we assume the plan we provide, with name of the streets, the distance to go and GPS directions, we assume it is enough to have a clear idea of where you should go. A preview of the monuments can be done with the pictures we saw above.

**Nearby places** Displaying the nearby places is also a classical feature, it can be interesting to implement it. For example, during the tour you organized, the user can realize that he is close to another building that he thought he won't have the time to visit, or maybe just he didn't thought about. It adds many possibility of unscheduled visits to the user, and will really improve the quality of the service offered by our application.

With those two application providers, we have a pretty complete overview of what tourism applications can contain, the essential features they should contain and what is useful to implement or not. We will now present our analyze of the Google Maps API.

## 3 Google Maps API

Google Maps allows us to display maps on their website. The Google Maps Application Programming Interface (Google Maps API) is a set of tools and methods used to include Google Maps' features in an application, and overlay our own data over them; the Google Maps API allows to customize maps and the information displayed on these maps.

The Google Maps API is composed of a wide array of specialized APIs :

- JavaScript API v3 : Embed an interactive map in a web page using Javascript;
- Maps for Work : Additionnal features to the standard Google Maps API devoted to business requirements and support;
- Google Maps SDK for iOS : Add an interactive map to an iOS application;

- Google Maps Android API : Add an interactive map to an Android application;
- Google Maps API Web Services : HTTP interfaces to Google services, allowing user to invoke Google Maps data through URL requests; this feature consists of a large set of other APIs, like the ones following;
- Google Maps Embed API : Embed an interactive map or street view panorama in a web page with HTTP requests;
- Google Places API : Display information about geographic locations, or points of interest; it can be used to give users contextual real-time information about their geographical position, and search for information about local businesses and monuments;
- Google Maps Image APIs : Set of APIs to embed a Google Maps static image or Street View imagery in a web page without JavaScript.

As we can see, the Google Maps API is rather complete and gives access to a large range of features, with specific APIs for each of these features. As part of our project, we will focus on the retrieving of information about locations using the Place API, creating an interactive map using the Maps Android API, and including functionalities and customized data to our application with Google API Web Services.

### 3.1 Google Maps Android API

**Add a Map to the application** The Google Maps API allows us to simply implement a map in our application through an XML layout file, and activity Java file. It provides several objects to organize the initial layout. At its initial state, our map does not contain any personalized content, tiles are empty, and icons are not yet clickable. The map object in the application is modeled by the *GoogleMap* class. This class handles the following operations :

- Connection to the Google Maps online services;
- Downloading map tiles;
- Displaying tiles on the screen;
- Displaying controls;
- Zoom.

It is possible to add our own behaviour in addition to the previous automatic operations, by providing new methods to the *GoogleMap* object. Within the UI, a map will be represented by either a *MapFragment* or *MapView* object.

The initial state of the map is entirely configurable : we can change the camera position, the map type, the available controls and gestures; this configuration is done using an XML layout file, or programmatically if the insertion of the map was done through the activity Java file.

**Drawing on the Map** This section provides all the tools we need to add customizable markers to our map. Markers have several properties that allow us to personalize their behaviour (draggable, icon, anchor, visible...). Listeners to marker-related events are provided by the API. Info windows can be anchored to a marker to display information on a location. Info Windows are customizable, and can be associated to an event if clicked.

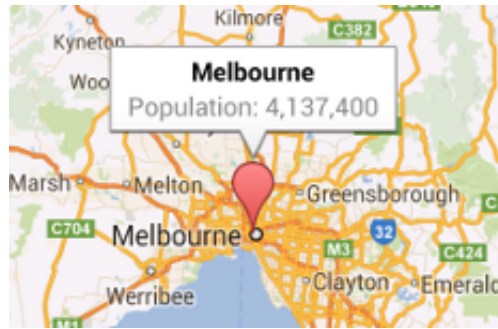


Figure 3: Google Maps marker and Info Window

The Google Maps API also offers to add shapes to the map, using *Polyline*, *Polygon* and *Circle* objects. As always, properties associated to these objects allow us to customize them. Finally, Ground and Tile overlays are images fixed to the map, allowing even more customization.

**Interaction with the Map** The API provides UI controls that allows users to interact with the map. There are built-in zoom controls, a compass, the "My location" button which focus the map on the current user's location (if localization is enabled), and a toolbar which displays icons that provide access to a map view or directions. Map gestures are enabled by default, like tilt, zoom, rotate and scroll. Events associated to gesture are programmable.

**Location Services** Thanks the GPS of the mobile device, and localization through mobile and wireless network, mobiles devices provide a set of localization technologies usable by our application. The Google Maps Android API provides the programming tools needed to benefit from those features. To add location awareness to our application, we must use the Google Play services Location API. This API lets us determine the location of the device, listen to location changes, and monitor geofences.

## 3.2 Google Maps API Web Services

The Google Maps API Web Services provides a large set of Web-oriented APIs using HTTP interfaces to Google services to provide geographic data to our application. We will only overview the APIs that seems the most relevant to the application we want to develop. All those APIs come with a large number of optional parameters to make them fit to our goal of use.

**Directions API** This service calculates directions between different locations, for static addresses. Dynamic direction calculation are provided in the JavaScript API.

**Distance Matrix API** This services provides travel distance and time between the elements of a matrix of origins and destinations.

**Geocoding API** The Geocoding API provides tools to convert addresses into geographical coordinates and vice versa.

## 3.3 Google Places API

The Google Place API provides a *PlacePicker* UI widget that allows our application's users to select a place on the map. It also helps to get the last known current location and return a list

of likely places for this location, and their likelihood. More importantly, this API retrieve and display informations about a place using the Google Play services.

## References

- [1] Bluebridge website. <http://gobluebridge.com/>. Accessed: 2015-04.
- [2] Google API developer documentation. <https://developers.google.com/maps/documentation/>. Accessed: 2015-04.
- [3] mtrip website. <http://www.mtrip.com/>. Accessed: 2015-04.