Travel Now - Approach Document

# Introduction

This document explains the steps involved in creating a phone gap application “Travel Now“, for the hackathon event, from scratch.

Explained below the different set ups and API calls required for the app.

# Required Setups

**Create Facebook App**

Register as a developer at developer.facebook.com. Create a new app in the app section in the developer portal.

**Install NodeJS**

Install from <http://nodejs.org/>

**Install Phonegap 3.1.0**

Ensure that you have NodeJS installed. Then open your commandline and run the following:

C:\> npm install -g phonegap

This will install the latest version of phone gap from the phone gap repository.

**Create a Phonegap Project**

Using command line, create phone gap project for Android and iOS

C:\> phonegap create //PATH\_OF\_FOLDER //IDENTIFIER //APP NAME

EX: phonegap create /Users/adminadmin/Desktop/THack/TravelNow com.amadeus.travelnow "TravelNow"

C:\> cd //PATH\_OF\_FOLDER

//iOS

C:\> phonegap local build ios

//Android

C:\> phonegap local build android

This creates the corresponding phone gap projects for iOS and Android platform.

**Install Java**

Install Java SDK (JDK and JRE) from <http://www.oracle.com/technetwork/java/javase/downloads/index.html> and complete the set up with setting up the PATH & CLASSPATH variables.

**Install Mongoose**

Download Mongoose from <https://code.google.com/p/mongoose/downloads/detail?name=mongoose-4.1.exe&can=2&q=>

**Install GitHub for Windows & Mac**

Download GitHubfrom <https://help.github.com/set-up-git-redirect>

**Install Open SSL & Keytool (available with JDK)**

Download Open SSL from <https://code.google.com/p/openssl-for-windows/downloads/detail?name=openssl-0.9.8e_X64.zip&can=2&q=> This is for generating hashkey for Android platform for Facebook SDK.

**Install ADT SDK & Android Emulator**

Install the ADT SDK from <http://developer.android.com/sdk/index.html#ExistingIDE>

**Install XCode & iOS Emulators**

Install XCode & Simulators from <https://developer.apple.com/xcode/>

# APIs

# Geo location

The *geolocation* object provides access to location data based on the device's GPS sensor or inferred from network signals.

*As of version 3.0, Cordova implements device-level APIs as plugins.* Using command line, add the corresponding phone gap plugin for Android and iOS

C:\> cd //PATH\_TO\_FOLDER

//iOS

C:\> phonegap local plugin add https://git-wip-us.apache.org/repos/asf/cordova-plugin-geolocation.git

//Android

C:\> phonegap local plugin add https://git-wip-us.apache.org/repos/asf/cordova-plugin-geolocation.git

**Methods**

# geolocation.getCurrentPosition

Returns the device's current position as a [Position](http://docs.phonegap.com/en/3.0.0/cordova_geolocation_geolocation.md.html#Position) object.

//Method implementation

navigator.[geolocation.getCurrentPosition](http://docs.phonegap.com/en/3.0.0/cordova_geolocation_geolocation.md.html#geolocation.getCurrentPosition)(onSuccess, onError);

**Response**

|  |  |
| --- | --- |
| **Parameter name** | **Description & Format** |
| latitude | Latitude in decimal degrees. (Number) |
| longitude | Longitude in decimal degrees. *(Number)* |
| altitude | Height of the position in meters above the ellipsoid. *(Number)* |
| accuracy | Accuracy level of the latitude and longitude coordinates in meters. *(Number)* |
| altitudeAccuracy | Accuracy level of the altitude coordinate in meters. *(Number)* |
| heading | Direction of travel, specified in degrees counting clockwise relative to the true north. *(Number)* |
| speed | Current ground speed of the device, specified in meters per second. *(Number)* |

**Sample Code**

Returns the device's current position as a [Position](http://docs.phonegap.com/en/3.0.0/cordova_geolocation_geolocation.md.html#Position) object.

// Display `Position` properties from the geolocation  
//

//  
var onSuccess = function(position) {  
    alert('Latitude: '          + position.coords.latitude          + '\n' +  
          'Longitude: '         + position.coords.longitude         + '\n' +  
          'Altitude: '          + position.coords.altitude          + '\n' +  
          'Accuracy: '          + position.coords.accuracy          + '\n' +  
          'Altitude Accuracy: ' + position.coords.altitudeAccuracy  + '\n' +  
          'Heading: '           + position.coords.heading           + '\n' +  
          'Speed: '             + position.coords.speed             + '\n' +  
          'Timestamp: '         + position.timestamp                + '\n');  
};

// Show an alert if there is a problem getting the geolocation  
    //  
    function onError() {  
        alert('onError!');  
    }

# EventFul API

This function handles event search. It performs the requested search, returning the results as an JSON response.

# Inspiration - Time of the Season

# URL

<http://1.glassy-song-375.appspot.com/GetTopEvents?date=FutureDate&page_size=100&keywords=(Festivals%20||%20Holiday)&sort_order=popularity&sort_order=ascending&max_cities=5>

http://1.glassy-song-375.appspot.com/GetTopEvents?date=FutureDate&page\_size=100&keywords=(Festivals%20||%20Holiday)&sort\_order=popularity&sort\_order=ascending&max\_cities=5&lat=12.9667&lon=12.9667

This call returns places/cities data, in JSON format, according to the most popular events, based on the inspiration/categories, occurring for the specified dates.

**Input Parameters**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Description** | **Sample Input** |
| authentication | This method controls access via [optional user authentication](http://api.eventful.com/docs/auth) | app\_key (handled at the server side) |
| keywords | The search keywords. (string) | Festivals, Health & Wellness, Holiday |
| location | A location name to use in filtering the search results. (string) (optional) | Locations in the form "San Diego", "San Diego, TX", "London, United Kingdom", and "Calgary, Alberta, Canada"  Also, Common geo coordinate formats ("32.746682, -117.162741") are also accepted, but the "within" parameter is **required** in order to set a search radius. |
| date | Limit this list of results to a date range, specified by label or exact range. (string) | Currently supported labels include: "All", "Future", "Past", "Today", "Last Week", "This Week", "Next week", months name e.g. "October"  Exact ranges can be specified the form 'YYYYMMDD00-YYYYMMDD00', for example '2012042500-2012042700'; the last two digits of each date in this format are ignored |
| category | Limit the search results to this category ID. (string) (optional) | A list of categories may be specified separated by commas. See [/categories/list](http://api.eventful.com/docs/categories/list) |
| within | If within is set and the "location" parameter resolves to a specific geolocation, the search will be restricted to the specified radius.   If the "location" parameter does not resolve to a specific location, this parameter is ignored. (integer) |  |
| units | One of "mi" or "km", the units of the "within" parameter. Defaults to "mi". (string) | "mi" or "km" |
| sort\_order | One of 'popularity', 'date', or 'relevance'. Default is 'relevance'. (string) | 'popularity', 'date', or 'relevance' |
| sort\_direction | One of 'ascending' or 'descending'. Default varies by sort\_order. (string) | 'ascending' or 'descending'. |
| page\_size | The desired number of results per page. (integer) |  |
| max\_cities | The desired number of cities required in the result. (integer) | max\_cities=5 |
| lat | The latitude received using the Geo location using GPS | lat=12.9667 |
| lng | The longitude received using the Geo location using GPS | lon=77.5667 |
| max\_events | Returns specified list of events for any destination city |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Parameter name** | **Description** | **Sample Input** |
| Cityname | Name of the city as the node of the JSON with the array list of events | "New York": [  {  "region\_name": "New York",  "performers": {  "performer": {  "id": "P0-001-000112433-3",  "linker": "evdb",  "short\_bio": "Scottish indie rock band",  "name": "Frightened Rabbit",  "url": "http://eventful.com/performers/frightened-rabbit-/P0-001-000112433-3?utm\_source=apis&utm\_medium=apim&utm\_campaign=apic",  "creator": "ioda"  }  },  "image": "http://s4.evcdn.com/images/medium/I0-001/003/293/375-3.jpeg\_/frightened-rabbit-75.jpeg",  "longitude": "-73.9891056",  "latitude": "40.7316190",  "event\_name": "Frightened Rabbit",  "country\_name": "United States"  } |
| region\_name |  | "region\_name": "New York", |
| longitude | Longitude of the place of the event | "longitude": "-73.9891056" |
| latitude | Latitude of the place of the event | "latitude": "40.7316190" |
| event\_name | Name of the event. | "event\_name": "Frightened Rabbit", |
| country\_name | Name of the country of the city. | "country\_name": "United States" |
| Performers | Array list of performers for the corresponding event | "performers": {  "performer": {  "id": "P0-001-000112433-3",  "linker": "evdb",  "short\_bio": "Scottish indie rock band",  "name": "Frightened Rabbit",  "url": "http://eventful.com/performers/frightened-rabbit-/P0-001-000112433-3?utm\_source=apis&utm\_medium=apim&utm\_campaign=apic",  "creator": "ioda"  } |
| image | URL of the image for the event | "image": "http://s4.evcdn.com/images/medium/I0-001/003/293/375-3.jpeg\_/frightened-rabbit-75.jpeg" |

|  |  |
| --- | --- |
| [**http://wikitravel.org/en/Las\_Vegas**](http://wikitravel.org/en/Las_Vegas) |  |
| **◄** | [**https://www.googleapis.com/freebase/v1/text/en/london**](https://www.googleapis.com/freebase/v1/text/en/london) | 3:37:23 PM |
| **◄** | [**https://www.googleapis.com/freebase/v1/topic/en/london?filter=/image&limit=1**](https://www.googleapis.com/freebase/v1/topic/en/london?filter=/image&limit=1) | 3:38:04 PM |
| **◄** | [**https://usercontent.googleapis.com/freebase/v1/image/m/04jpl**](https://usercontent.googleapis.com/freebase/v1/image/m/04jpl) | 3:38:26 PM |