GEOCADE

Location Aware Leaderboard API Implementation Guide

Introduction

Disclaimer of Warranty

Geocade Inc. makes no representations or warranties, either expressed or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Overview

The Geocade location aware leaderboard API provides integration into services that track scores and ranking in gaming communities based on a player's location. Each game integrated into the Geocade gaming community comes enabled with gaming communities in each city, town and state in the United States as well as most international locations.

If you would like to have only a global leaderboard you may pass in 0 for lat and 0 for lng. We do encourage use of of the location portion of Geocade as we have found that global leaderboards get boring for all but the top players and we can also get better CPM's for you if we have location information.

Definitions, Acronyms, and Abbreviations

Token

Unique token assigned to a client by Geocade Inc.

GPS Body Paint token is: 51460djgsdp30ajsjb9shvfsd38an

Game ID

Unique game id assigned to a client by Geocade Inc.

GPS Body Paint game_id is: 51460

Base URL

Constant portion of HTTP call to the Geocade Leaderboard API service.

The base url for the staging server is `staging.geocade.com`

Paramenter(s)

Variable name and potential value(s) that can be appended to the Base URL

API Calls

1. submitScore

Returns an html page to be displayed in a web view that contains:

- the rank of the score submitted within the given gaming community
- the top ten scores in the gaming community
- links to high score state (or region) and national leaderboards
- an advertisement (which may be national or local)

Base URL:
http://staging.geocade.com/scores/submitScore

Parameters Definition:

- token **(required)** – token id
- total **(required)** - the achieved score of the user
- gid **(required)** - the unique game id
- lat **(required)** – the current latitude of the user. Used to determine the correct gaming community
- lng **(required)** – the current longitude of the user. Used to determine the correct gaming community
- uid **(required)** - the unique device id for the user. This **must** be the iPhone/Touch device id
- msg **(optional)** – additional information about the score (example -used to indicate level achieved).
- cid **(required in production) –** To prevent score fraud we are supplying a checksum formula below to verify score submissions.

Checksum formula:

Games using money based scoring systems must multple their score by 100 to generate a whole number prior to using the formula below.

```
int md = 13, remainder, checksum;
remainder =  (score +  gameid) % md;
checksum = (remainder + 7)  * (int)abs(lat) * (int)abs(lng);
```

If the lat and lng are 0 or location cannot be found:

```
int md = 13, remainder, checksum;
remainder =  (score +  gameid) % md;
checksum = (remainder + 7)  * 42 * 71
```

For ease of integration and for testing purposes the checksum can be omitted or incorrect on the staging server.

You will find a checksum validator at the bottom of all the submitScore page. This will allow you to check to see if the checksum you generated matches the Geocade checksum. This is available on the integration server for troubleshooting purposes. If checksums do not match in production then the score will be rejected.

Example code for device id:

```
(NSString*)getDeviceId
{
  UIDevice *device = [UIDevice currentDevice];
  return device.uniqueIdentifier;
}
```

Example code for post:

```
(void)goToURL:(NSString*)urlString
{

//split fully formed url (urlString) into server/path(base) and
//parameters(body)

NSArray *urlList = [urlString componentsSeparatedByString:@"?"];

NSString *urlBase = [urlList objectAtIndex:0];
NSString *urlBody = [urlList objectAtIndex:1];

//NSString *myRequestString = @"id=1";
NSData *myRequestData = [ NSData dataWithBytes: [urlBody UTF8String ]
length: [urlBody length ] ];

 NSMutableURLRequest *request = [ [ NSMutableURLRequest alloc ]
  initWithURL: [ NSURL URLWithString:urlBase ] ];
               [ request setHTTPMethod: @"POST" ];
               [ request setHTTPBody: myRequestData ];
               [ request setValue:@"application/x-www-form-urlencoded"
forHTTPHeaderField:@"content-type"];

      [_webView loadRequest:request];

}
```

Example URL:

http://staging.geocade.com/scores/submitScore?lat=42.362097&lng=-71.082896&uid=gbpwef23plk71547a1a2c5ade7549c9decc209964b4ec&total=73&gid=**51460**&token=**51460djgsdp30ajsjb9shvfsd38an**

Notes:

- Upon the submission of a users first score they will directed to create a profile.
- The html returned by Geocade must be displayed in a web view and must not contain any attempt to block advertising.
- Navigation controls on the web view shall be limited to a button that returns the users to the game and an optional back button.
- The leaderboard colors may be customized including background, alternating row colors and table header color.
- We require that requests be made via http Post.

2. profile

Returns and html page to be displayed in a web view that contains:

- links to modify profile
- links to track scores and ranking across communities.

Base URL:
http://staging.geocade.com/users/profile

Parameters Definition:

- token **(required)** - token id
- gid **(required)** - the unique game id.
- id **(required)** - the unique device id for the user. This **must** be the iPhone/Touch device id

Example URL:
http://staging.geocade.com/users/profile?id=gbpwef23plk71547a1a2c5ade7549c9decc209964b4ec&gid=**51460**&token=**51460djgsdp30ajsjb9shvfsd38an**

Profile fields – entered by user in web view

- User name - (required and unique)
- Email - (optional) and not provided or sold to any external sources and only used to award prizes and for future alert notification features.
- Hometown -(optional) used only in providing the from location when reporting scores in the community
- State - (optional) used only in providing the from location when reporting scores in the community

Note:

- We require that requests be made via http Post.

Location Gathering

For the best possible result we recommend you obtain the location of the user during the game rather than just before submitting the score. Getting the location just before will result in a less reliable result.

Score Fraud and Profanity

Geocade provides profanity filters for user names and performs automated checks for score fraud (in addition to checksum validation). Upon request Geocade can change/remove scores and users from the leaderboards. Requests shall be made to support@geocade.com

Lowest Possible Score: TBD
Highest Possible Score: TBD

Integration Server

You are using an integration server. The data on the integration server can be flushed upon request.

WebView Sample Code

```
//
//  LocationTestViewController.m
//  LocationTest
//


#import "HighScoreViewController.h"

@implementation HighScoreViewController
@synthesize webView;
@synthesize leaderboardView;



/*
// The designated initializer. Override to perform setup that is required before the view is
loaded.
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
{
  if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
      // Custom initialization
  }
  return self;
}
*/

/*
// Implement loadView to create a view hierarchy programmatically, without using a nib.
- (void)loadView {
}
*/



// Implement viewDidLoad to do additional setup after loading the view, typically from a
nib.
- (void)viewDidLoad {
        webView.delegate = self;
  [super viewDidLoad];
        [self.view addSubview:leaderboardView];
        locationManager = [[CLLocationManager alloc] init];
        [locationManager setDelegate:self];
        [locationManager setDesiredAccuracy:kCLLocationAccuracyHundredMeters];
}
```

```
- (void)submitScore:(int)aScore
{
        [webView loadHTMLString:@"<center><h1>Submitting
Score…</h1></center>" baseURL:nil];
        score = aScore;
        [locationManager startUpdatingLocation];
}

- (void)locationManager:(CLLocationManager *)locManager
didUpdateToLocation:(CLLocation *)newLocation fromLocation:(CLLocation
*)oldLocation
{
        NSLog(@"lat: %f, lon: %f", newLocation.coordinate.latitude,
newLocation.coordinate.longitude);
        int gameid = GAMEIDHERE;
        float lat = newLocation.coordinate.latitude;
        float lng = newLocation.coordinate.longitude;
        UIDevice *device = [UIDevice currentDevice];
        NSString *uid = [device uniqueIdentifier];
        int md = 13, remainder;
        remainder =  (score +  gameid) % md;
        int cid = (remainder + 7)  * (int)abs(lat) * (int)abs(lng);

        NSString *urlString = [NSString
stringWithFormat:@"http://staging.geocade.com/scores/submitScore?token=TOKENHE
RE&gid=%i&lat=%f&lng=%f&uid=%@&total=%i&cid=%i", gameid, lat, lng, uid,
score, cid];
        NSLog(urlString);
        //split fully formed url (urlString) into server/path(base) and parameters(body)
        NSArray *urlList = [urlString componentsSeparatedByString:@"?"];
        NSString *urlBase = [urlList objectAtIndex:0];
        NSString *urlBody = [urlList objectAtIndex:1];

        NSData *myRequestData = [ NSData dataWithBytes: [urlBody UTF8String ]
length: [urlBody length ] ];
        NSMutableURLRequest *request = [ [ NSMutableURLRequest alloc ]
initWithURL: [ NSURL URLWithString:urlBase ] ];

        [ request setHTTPMethod: @"POST" ];
        [ request setHTTPBody: myRequestData ];
        [ request setValue:@"application/x-www-form-urlencoded"
forHTTPHeaderField:@"content-type"];
        [webView loadRequest:request];
        [locationManager stopUpdatingLocation];
}
```

```
- (IBAction)onReturn:(id)sender
{
        [self.view removeFromSuperview];
}

- (void)webView:(UIWebView *)aWebView didFailLoadWithError:(NSError *)error
{

        if ([error code] != -999) {

        [webView loadHTMLString:@"<center><h1>Cannot contact high score
server.</h1></center>" baseURL:nil];

}

// Override to allow orientations other than the default portrait orientation.
-
(BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrient
ation {
  // Return YES for supported orientations
  return (interfaceOrientation == UIInterfaceOrientationPortrait);
}



- (void)didReceiveMemoryWarning {
  [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview
  // Release anything that's not essential, such as cached data
}


- (void)dealloc {
        [locationManager release];
        [webView release];
        [leaderboardView release];
  [super dealloc];
}

@end
```