Trimaps SDK

Javascript Integration Guide

Introduction

This document describes how to integrate the Trimaps SDK to your web/mobile web application though the Javascript API.

Trimaps SDK allows application developers to draw a user's location on a custom map image, based on latitude/longitude coordinates provided by the device's GPS.

Before you start, make sure you have downloaded the Trimaps Javascript SDK (trimaps-sdk-js-0.5.zip) from trimaps.com and that you have generated the Trimaps Localizer File (.tlf) for your project map.

Integration of Trimaps SDK

1. **Make the SDK Library available to your webpages**

Unzip trimaps-sdk-js-0.5.zip and place its content in a directory served by the web server, referred later as <path>.

2. **Import the Javascript library**

In the HTML page where you plan to use the Trimaps SDK, import the Javascript library.

```
<script type="text/javascript" language="javascript"
src="<path>/com.tribab.trimaps.JSLibrary.nocache.js"></script>
```

3. **Invoking Trimaps SDK**

Invoking the SDK is a straightforward process from your Javascript code.

- load the map Localiser File content using the `loadMap` function

  ```
  /* loads a map, given its file content */

  function loadMap(fileContent);
  ```

  If the file content can not be parsed by the library, an Alert is thrown.

- call the function `getPositionOnCustomMap` to retrieve the (x,y) location of a point on your custom map based on a GPS provided (latitude, longitude).

  ```
  /* returns a table of integer (x,y) */

  function getPositionOnCustomMap(latitude, longitue);
  ```

  The returned array contains the (x,y) coordinates at zoom level 1 (with respect to the original image size loaded on trimaps.com).

4. **Unit Testing**

Unit testing the integration of the Trimaps SDK library in your webpage can be easily achieved

with standard browser's developer tools.

Advanced features

- Handling several maps

Some customers express the need to handle several maps in their application. The Trimaps SDK library handles this with the notion of *Context.* A context tells Trimaps which map has to be considered active when the API is invoked. On a context switch, Trimaps in turn will load all the mapping data to be able to process the requests on the active context. Trimaps keeps a context active until the next context switch call.

Once a context has been made active in the library, the mapping data remains loaded so that a subsequent context switch will not require a re-load of the first context.

When using the library with only one map, the calling application does not need to handle the notion of context: since there is only one, Trimaps knows obviously which one to use.

However, with several maps, the calling application must explicitly tell Trimaps which context has to be used :

- when loading a map, specify which context name must be associated with it

```
/* Loads a map and associates a context with it.*/
loadMapInContext(context, fileContent);
```

- before you invoke `getPositionOnCustomMap`, make sure the correct context is the active one by calling `setContext`

```
/* Sets the given context active
 * returns true if the context was loaded; false otherwise */
setContext(context);
```

Did you know ?

Beside **Trimaps SDK** for Javascript, did you know that Trimaps offers additional services ?

- **Trimaps SDK** for Java/Android & iOS
- HTML **Map Viewer**
- **Trimaps EASY** to quickly publish any map image online
- Trimaps offers additional services for all your map related projects.

  Do not hesitate to contact us for more info about those services: info@trimaps.com

Support & more info

Trimaps service support can be reached through trimaps.com, or at support@trimaps.com

http://www.trimaps.com