**Developer Guide**

**jWebSocket**

**MapSensors**

**1.0**

**Version history?**

# Introduction

This manual describes how the developer can modify and use the application code of MapSensors developed with the jWebSocket framework. This document has been created for software developers to create, assemble and deploy applications using the jWebSocket framework.

MapSensors location is an application that allows users to view a map in real time via the mobile Web, in addition to reference the current position of the mobile device. It differs from other applications of location because it works in real time through the new WebSocket protocol that makes it possible. It was developed using the framework jWebSocket, allowing the user to view the potential for locating jWebSocket in real time. As mention already for the user guide: The main benefit is to distribute the location data in realtime, please expose that more clearly.

# Infrastructure of solution

* The MapSensors application consumes features from the JavaScript libraries PhoneGap and OpenLayers. Please provide download links.
* The MapSensors application receives and sends data through the jWebSocket client
* The server contains the jWebSocket plugin the application that is responsible for managing the maps with the map server. Which Plugin, please give the name.
* **2. Requirements and prerequisites**
* Resource requirements
  + Smartphone with OS Android 2.3.3 or higher.
  + WIFI Wireless Access Point.
  + What about GSM geolocation?
* Equipping
  + SDK android 2.3.3, where to get, how to use, to install?
  + Plugin ADT-12, where to get, how to use, to install?
  + GeoServer map server, where to get, how to use, to install?
  + Eclipse Helios, where to get, how to use, to install?

# Modularization of the application

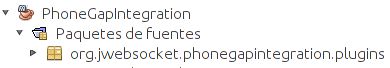
* 1. **The client side**

Please give at least an introductional sentence like “the following table shows….”

|  |  |
| --- | --- |
| Nombredelproyecto: | MapSensors |
| LocalizacióndelasfuentesenelservidorSVN: | http://repo.hab.uci.cu/svn/tesis/Tercer\_Corte\_de\_Tesis/JWS/Carlos\_Karen/Codigo/ |
| SVNbranch: | jWebSocket-1.0 |
| Módulo: | MapSensors.apk |
| Estructura de los paquetes |  |
| Phonegap-1.1.0.js  Libreri JavaScript of PhonGap | |
| Main.js:  Contains the definition and implementation of the application of maps and sensors. | |
| Base64:  Contains the libraries responsible for converting a String to an image in base64. | |
| OpenLayers:  Contains the definition and implementation of the library Openlayers | |
| Websocket:  Contains the implementation of the classes responsible for performing communication with the server. | |
| Master.css , index.html:  Containing the definition of the application for the control of the interface. | |

* 1. Please translate all above properly to English and correct typos!
  2. **The Server Side**

The server has a jWebSocket plugin receiving requests from users and performs MapSensors request to Geoserver server to respond to users. (This is not explained in the Administrator Guide?) This server is structured as follows.

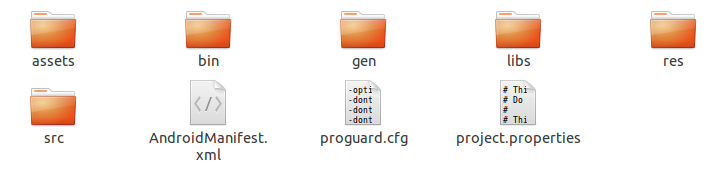


Already integrated into our structure? What is “a plug-in” ¿

Here I expect far more details. Also in the Administrator Guide.

# Source code structure

# Client:



# Description of the elements:

assets: This directory stores files that are part of the solution of the source code for the web, html files, js and css. Hmm, what files? Purpose?

bin: In this directory the executables, classes? file is generated. Apk which is the application executable.

libs: Directory where you copied. Jar to be loaded by the application. What .jar(s)? Give example(s).

src: This directory stores files with source code (. Java) that are part of the solution

AndroidManifest.xml: Configuration file of the application.

gen, res, proguard.cfg, proyect.properties: Folders and files are automatically created when you build a NetBeans project.

# Libraries and tools used

**On the client side:**

The libraries and classes for the extension MapSensors clients are available in Java and JavaScript files. The requirements of external libraries at the client side are:

Give some introductional sentence, what is OpenLayers and PhoneGap about.

|  |  |
| --- | --- |
| **Library** | **Description** |
| OpenLayers | Version 2.1.1 |
| Phonegap | Version 1.1.0 |

Give some links where to download, where to get more information.

# Standard code

View Template Code Standard:

[http://repo.hab.uci.cu/svn/tesis/JWS/Carlos\_Karen\_Cespedes/artefactos\_arreglados/Plantilla Estandar de codigo.doc](http://repo.hab.uci.cu/svn/tesis/JWS/Carlos_Karen_Cespedes/artefactos_arreglados/Plantilla%20Estandar%20de%20codigo.doc)

I can’t access that, can you please submit that to our SVN, if publically available.

# Libraries and Tools used

* Framework PhoneGap, MIT License.
* Framework OpenLayers, the BSD license
* SDK Android 2.3.3
* Plugin ADT-12

# Hardware requirements

To use the application for location management in mobile web, developed with the jWebSocket framework you must have the following features in your work environment:

* Smartphone with operating system Android 2.3.3 and higher ?

# Network protocol used

The application of location through the mobile web with the framework developed jWebSocket, used to establish the protocol comunicaciónel WebSocket technology that provides a bidirectional communication channel and full-duplex over a single TCP socket.

# Detailed description of the application code

**11.1 Server:**

The basic methods in the plugin on the server are:

ProcessToken: Receive incoming requests from any client WebSocket

getURLData: Get the url of an image, performs the request and returns it converted to a String.

**Client:**

The methods on the client side that describe the most important features are:

openWebSocket: Set the connection to the server.

map: Allows load map for proper display. A bit more details here please.

moveFeature: Moves the given feature in the given map.

rotateFeature: Rotates a given feature

toggleAcceleration: Capture the event of acceleration sensor

getLocation: Subscribes to the events of gps.