



Hotcity

## Hotcity API User's Manual

Version: 1.1 - Final  
August 2008

## TABLE OF CONTENT

1	INTRODUCTION.....	4
2	TECHNOLOGY.....	5
2.1	PHP.....	6
2.2	JAVA.....	8
2.1	.NET.....	12
3	WEB SERVICE PARAMETERS.....	15
3.1	DATA TYPES .....	15
3.2	DATA STRUCTURES.....	15
3.2.1	<i>PortailDeviceInfo</i> .....	15
3.2.2	<i>UserInfo</i> .....	16
3.2.3	<i>Position</i> .....	16
3.2.4	<i>PositionXY</i> .....	16
3.2.5	<i>UserAp</i> .....	17
3.2.6	<i>UserApXY</i> .....	17
3.2.7	<i>HistDist</i> .....	17
4	AVAILABLE METHODS.....	18
4.1	DEVICE CHARACTERISTICS.....	18
4.1.1	<i>getBrowserType</i> .....	18
4.1.2	<i>getDeviceInfo</i> .....	18
4.1.3	<i>getDeviceInfoEx</i> .....	19
4.1.4	<i>GetOsType</i> .....	19
4.1.5	<i>GetScreenSize</i> .....	20
4.1.6	<i>setDeviceInfo</i> .....	20
4.1.7	<i>setDeviceInfoEx</i> .....	21
4.1.8	<i>setPreferredLanguage</i> .....	21
4.1.9	<i>setScreenSize</i> .....	22
4.2	USER PROFILE INFORMATION.....	23
4.2.1	<i>SetUserProfile</i> .....	23
4.2.2	<i>getUserProfile</i> .....	23
4.3	WURFL CAPABILITIES.....	24
4.3.1	<i>canFlashLite10</i> .....	24
4.3.2	<i>canFlashLite11</i> .....	24
4.3.3	<i>canHtmlWeb32</i> .....	24
4.3.4	<i>canHtmlWeb40</i> .....	25
4.3.5	<i>canVideo3GPP</i> .....	25

4.3.6	<i>canVideo3GPP2</i> .....	25
4.3.7	<i>canVideoMOV</i> .....	26
4.3.8	<i>canVideoMP4</i> .....	26
4.3.9	<i>canVideoRealMedia10</i> .....	26
4.3.10	<i>canVideoRealMedia8</i> .....	27
4.3.11	<i>canVideoRealMedia9</i> .....	27
4.3.12	<i>canVideoWMV</i> .....	27
4.3.13	<i>canXhtmlBasic</i> .....	28
4.3.14	<i>hasHttpsSupport</i> .....	28
4.3.15	<i>hasPointingDevice</i> .....	28
4.3.16	<i>hasVideoSupport</i> .....	29
4.3.17	<i>maxImageHeight</i> .....	29
4.3.18	<i>maxImageWidth</i> .....	29
4.3.19	<i>uaProf</i> .....	30
4.4	IDENTIFICATION.....	30
4.4.1	<i>GetMAC</i> .....	30
4.5	USERS.....	31
4.5.1	<i>Authenticate</i> .....	31
4.5.2	<i>GetUserById</i> .....	31
4.5.3	<i>getUserByMacAddress</i> .....	32
4.6	LOCATION.....	32
4.6.1	<i>getApPosition</i> .....	32
4.6.2	<i>getApPositionXY</i> .....	32
4.6.3	<i>getUserAp</i> .....	33
4.6.4	<i>getUserApXY</i> .....	33
4.6.5	<i>getUserDistanceFromAp</i> .....	34
4.6.6	<i>getUserHistoryAp</i> .....	34
4.6.7	<i>getUserHistoryApFromDate</i> .....	35
4.6.8	<i>getUserKnownAp</i> .....	35
4.6.9	<i>getUserPosition</i> .....	36
4.6.10	<i>getUserPositionXY</i> .....	36
4.7	MAPS.....	37
4.7.1	<i>getMap</i> .....	37
4.7.2	<i>getMapXY</i> .....	37

## **1 Introduction**

This document presents the API to be used by applications running on the HotCity environment.

This API offers applications the capability to easily communicate with the HotCity platform, using Web Services.

The API gives access to various functions with, for example, the opportunity to localize the user, to get information about device's capability, ...

## 2 Technology

Scope of the API : offer services to various software platforms. Web services technology is used to handle communications. Services are described in a standard WSDL file available at:

<http://hotcity.lu:8095/LBS/LbsWebService?wsdl>

Applications use this XML file to get available methods and give details about parameter types, result types and exceptions raised.

The following sections give an overview to create clients using PHP, Java and .NET. For more information on writing web services client, refer to the specific documents.

## 2.1 PHP

The following example, written in PHP, shows how to realize:

- storage of a user profile attached to a user's device,
- authenticate a user against the platform,
- search user's information based on its userID or its MAC address.

```
<?php
// New SOAP Client
try {

    ini_set("soap.wsdl_cache_enabled", "0");

    $wservice = "http://hotcity.lu:8095/LBS/LbsWebService?wsdl";

    $client = new SoapClient($wservice, array('trace' => 1));

    $userName = "username";
    $pass = "password";
```

```
//on set device info
echo "***** setdevice info *****<br>";
$res = $client->setDeviceInfo(array('macAddress'=> '00:40:96:ac:47:02', 'userAgent'=>
$_SERVER['HTTP_USER_AGENT']));

echo "***** getdevice info *****<br>";
$res = $client->getDeviceInfo(array('macAddress'=> '00:40:96:ac:47:02'));
    var_export($res);

// authenticate
echo "***** authenticate *****<br>";
$res = $client->authenticate(array('userName'=>$usrName,
    'password'=>$pass, 'applicationKey'=>'test'));
    var_export($res);
print "<br>\n";
// getuserbyid
echo "***** getuserbyid *****<br>";
$res = $client->getUserById(array('userId'=>$res->return, 'applicationKey'=>'test'));
var_export($res);
// getuserbyMac
echo "***** getuserbyMacAdress *****<br>";
$res = $client->getUserByMacAddress(
    array('macAddress'=>'00:40:96:ac:47:02', 'applicationKey'=>'test'));
var_export($res);

print "<pre>\n";
print "<pre>\n";
print "Request: \n".htmlspecialchars($client->__getLastRequest())."\n";
print "Response: \n".htmlspecialchars($client->__getLastResponse())."\n";
print "</pre>";
} catch (SoapFault $fault) {
    echo $fault;
}
?>
```

## 2.2 Java

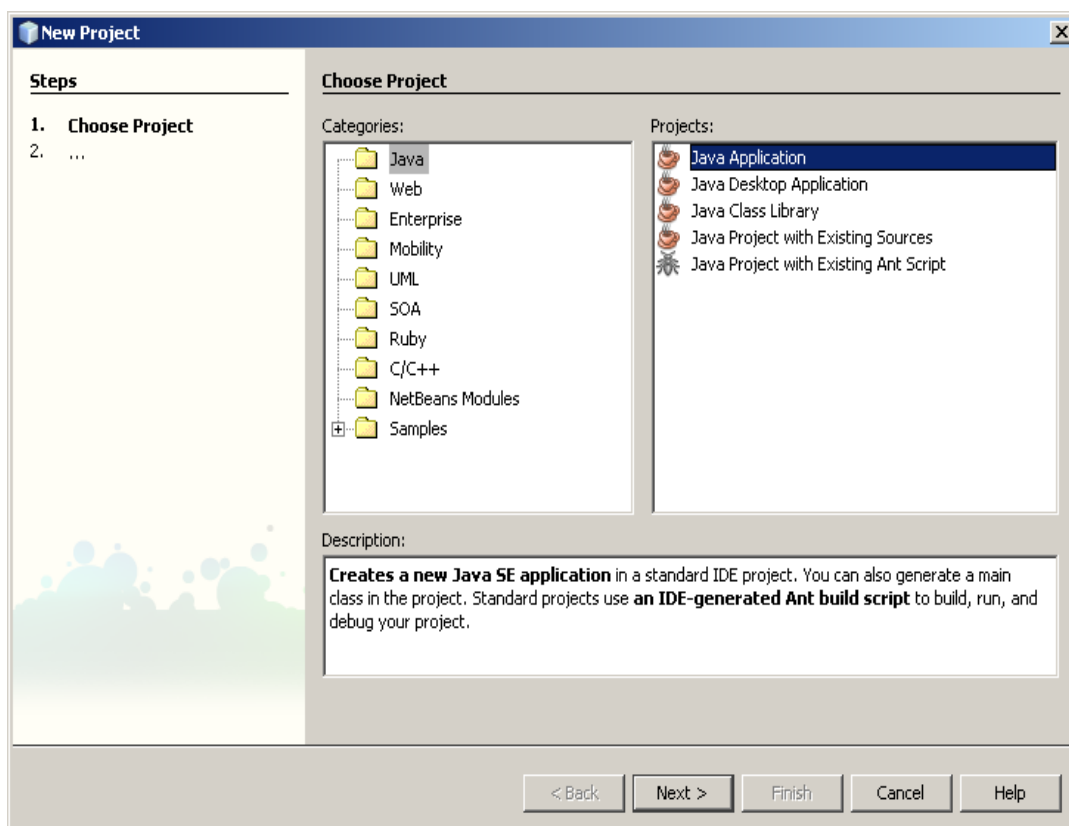
Various Integrated Development Environment (IDE) ease producing the required Java code to call a Web Service. Two IDE are currently very convenient and free.

First one, "Eclipse", is sponsored by IBM. More information about Eclipse is available at <http://www.eclipse.org/>.

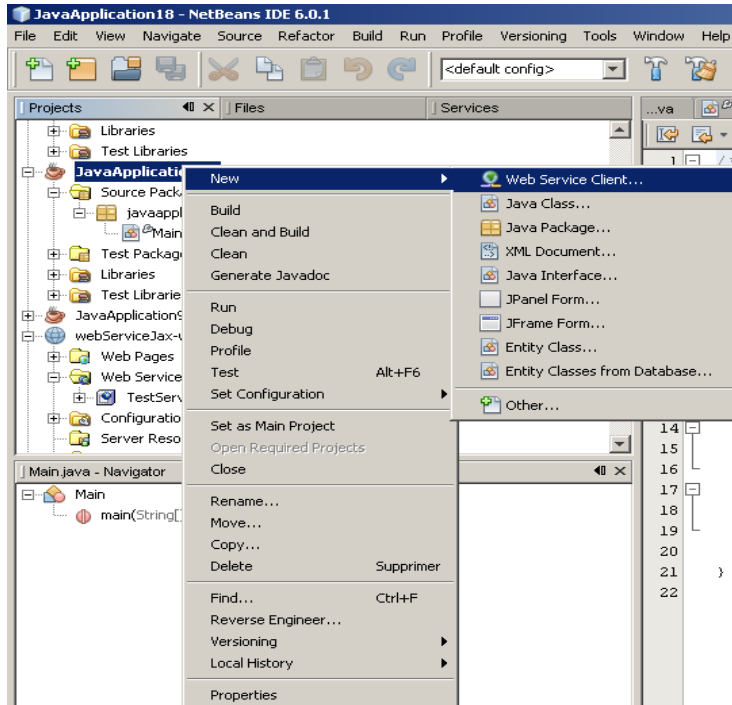
Second one, Netbeans, is sponsored by Sun. More information about Netbeans is available at <http://www.netbeans.org/>.

Examples below have been made using Netbeans; other IDE can be used in a similar way.

In order to create a Web Service Client, create a new Java project.

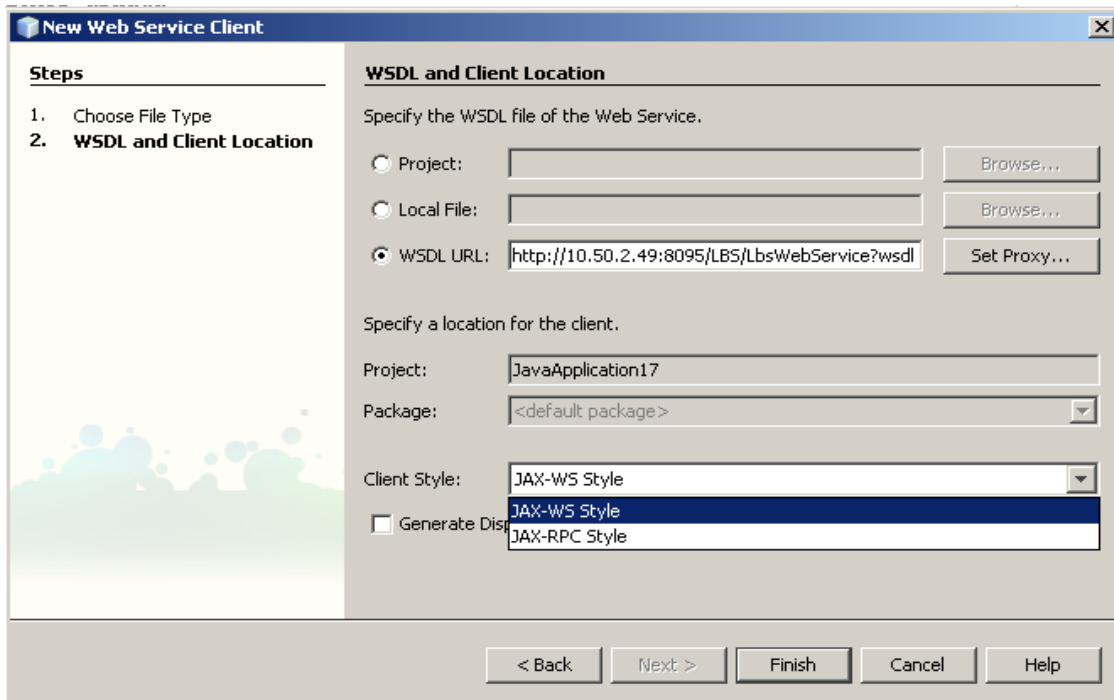


Then, create a new “Web Service Client” in this project.



The Web Service Client's properties will be set in the next screen (see below):

- set the path to the WSDL file: <http://hotcity.lu:8095/LBS/LbsWebService?wsdl>
- select in which package the classes (“stubs”) will be generated
- select the Web Service Client type: JAX-WS or JAX-RPC.



The code sample below describes how to access the Web Service.

This example uses the methods “authenticate”, “getUserById”, “getUserByMacAddress”:

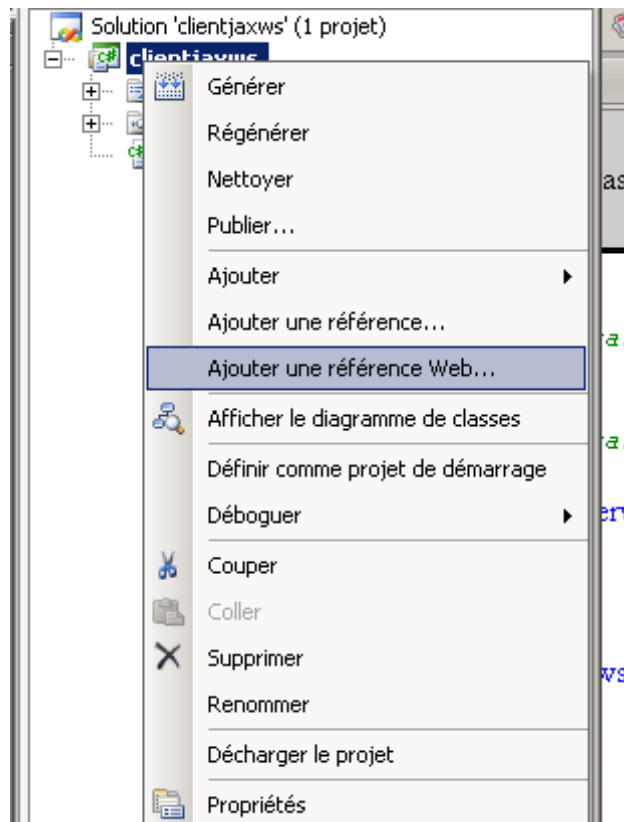
```
package telbsremote;

public class Main {
    public static void main(String[] args) {
        try {
            lu.pbs.hotcity.lbs.ws.LbsWebServiceService service = new lu.pbs.hotcity.lbs.ws.LbsWebServiceService();
            lu.pbs.hotcity.lbs.ws.LbsWebService port = service.getLbsWebServicePort();
            String user = "tempuser";
            String pass = "tempuser";

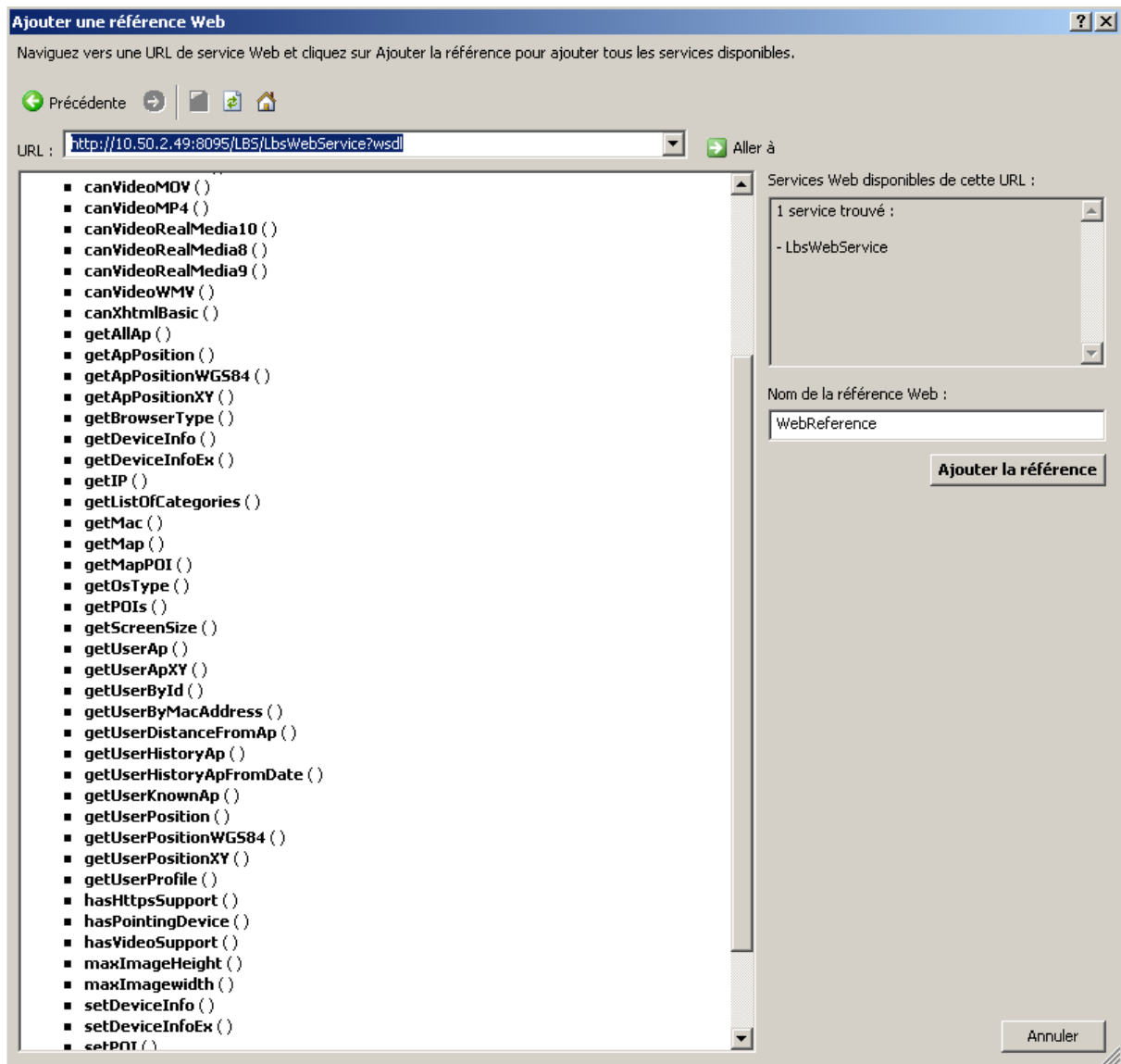
            long r = port.authenticate(user, pass, "test");
            int userId = Integer.parseInt(r + "");
            java.lang.String securityKey = "test";
            lu.pbs.hotcity.lbs.ws.PortailMember resultid = port.getUserById(userId, securityKey);
            System.out.println("Resultid = " + resultid.getFirstName() + " " + resultid.getLastName());
            lu.pbs.hotcity.lbs.ws.PortailMember result = port.getUserByMacAddress("00:00:00:ac:47:02",
securityKey);
            System.out.println("Result = " + result.getFirstName() + " " + result.getLastName());
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

## 2.1 .NET

To create a Web Service Client using .Net with Visual Studio 2005, create a new C# project and create a new Web reference within this project.



Then, set the path to the WSDL file: <http://hotcity.lu:8095/LBS/LbsWebService?wsdl> .



The code sample below introduces how to access the Web Service.

This example uses the methods “authenticate”, “getUserById”, “getUserByMacAddress”:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Web.Services.Protocols;
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {

            String password="tempuser";
            String username="tempuser";

            Project2.WebReference.LbsWebServiceService service = new
Project2.WebReference.LbsWebServiceService();
            long r = service.authenticate(username, password,"test");
            Project2.WebReference.portailMember member = service.getUserById(r, "test");
            System.Console.WriteLine(member.lastName);
            Project2.WebReference.portailMember member2 =
service.getUserByMacAddress("00:00:00:ac:47:02","test");
            System.Console.WriteLine(member2.userName);

        }
    }
}
```

### 3 Web Service parameters

#### 3.1 Data types

String	Character strings - all string literals in Java programs, such as "abc", are implemented as instances of this class.
Int	The int data type is a 32-bit signed two's complement integer - it has a minimum value of -2,147,483,648 and a maximum value of 2,147,483,647 (inclusive).
Float	Wraps a value of primitive type float - the float data type is a single-precision 32-bit IEEE 754 floating point.
Double	Wraps a value of primitive type double - the double data type is a double-precision 64-bit IEEE 754 floating point.
Boolean	Wraps a value of primitive type boolean - the boolean data type has only two possible values: true and false.
Date	Represents a specific instant in time, with millisecond precision.

#### 3.2 Data structures

Some information is transmitted to and from the Web Service using data structures – they are detailed hereafter.

##### 3.2.1 PortailDeviceInfo

Contains the description of a device.

String MAC	MAC address of the device.
Date lastUpdate	Date of the last update.
String preferredLanguage	User's preferred language as defined by the standard ISO 639-1.
String largeScreen	Screen size of the device (possible values are "mini" or "maxi" ).
String osType	Operating system of the device.
String browserName	Browser used to access the service.
String browserVersion	Version of the browser used to access the service.
String userAgent	Client application which provided the device information.

### **3.2.2 UserInfo**

Provides information about the current user.

Long id	User's identification number on the platform.
String firstName	User's first name.
String lastName	User's last name.
String userName	Name used to authenticate the user and to create his(her) session.
String email	User's email address.

### **3.2.3 Position**

Provides information about a user's location or an Access Point's location - information is provided according to the World Geodetic System reference frame WGS84 (longitude, latitude).

Double longitude	Longitude.
Double latitude	Latitude.
String apMacAddress	MAC address of the Access Point on which the current user is connected.

### **3.2.4 PositionXY**

Provides information about a user's location or an Access Point's location - information is provided according to the LUREF projection system (X,Y).

Double X	Location value on the X axis.
Double Y	Location value on the Y axis.
String apMacAddress	MAC address of the Access Point on which the current user is connected.

### **3.2.5 UserAp**

Provides information about an Access Point according to the World Geodetic System reference frame WGS84 (longitude, latitude).

String macAddress	MAC address of the Access Point.
Long userApId	Identification number of the Access Point.
Double longitude	Longitude of the Access Point.
Double latitude	Latitude of the Access Point.

### **3.2.6 UserApXY**

Provides information about an Access Point according to the LUREF projection system (X,Y).

String macAddress	MAC address of the Access Point.
Long userApId	Identification number of the Access Point.
Double X	Location of the Access Point on the X axis according to the LUREF projection system.
Double Y	Location of the Access Point on the Y axis according to the LUREF projection system.

### **3.2.7 HistDist**

Provides chronological records of distances to Access Points.

Long Id	Identification number of the Access Point.
Double dist	Distances to the Access Point.

## 4 Available methods

### 4.1 Device characteristics

#### 4.1.1 *getBrowserType*

**Definition**

Returns the type of browser used on the user's device.

**Input parameters**

String *macAddress*: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

**Returns**

String: Browser type.

**Exception**

If a runtime error occurs, an exception `"lu.pbs.hotcity.lbs.exception.LbsException"` is raised.

#### 4.1.2 *getDeviceInfo*

**Definition**

Returns various device information, based on its MAC address.

Prior to calling this method, the `"setDeviceInfo"` method must have been invoked.

**Input parameters**

String *macAddress*: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

**Returns**

`PortailDeviceInfo`: Device characteristics.

**Exception**

If a runtime error occurs, an exception `"lu.pbs.hotcity.lbs.exception.LbsException"` is raised.

### **4.1.3 *getDeviceInfoEx***

#### **Definition**

Returns various device information, based on its IP address.

Prior to calling this method, the "setDeviceInfo" method must have been invoked.

#### **Input parameters**

String ipAddress: IP address of the device.

#### **Returns**

PortailDeviceInfo: Device characteristics.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.1.4 *GetOsType***

#### **Definition**

Returns the operating system of the device.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

#### **Returns**

String: Operating system information.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.1.5 GetScreenSize**

#### **Definition**

Returns the screen size of the device.

Prior to calling this method, the "setDeviceInfo" method must have been invoked.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

#### **Returns**

String: Screen size of the device - possible values:

- "maxi" if the device is a laptop,
- "mini" if the device is cell phone (palm, pocket pc ...).

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.1.6 setDeviceInfo**

#### **Definition**

Reads the device characteristics from the device itself and stores them in the database.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

String userAgent: User agent of the device.

#### **Returns**

Boolean: Possible values are "true" if the device characteristics have been successfully read and stored, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.1.7 setDeviceInfoEx**

#### **Definition**

Reads the device characteristics from the device itself and stores them in the database - this method also returns the MAC address of the device, based on an IP address.

#### **Input parameters**

String ipAddress: IP address of the device.

String userAgent: User agent of the device.

#### **Returns**

String: MAC address of the device.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.1.8 setPreferredLanguage**

#### **Definition**

Sets the user's preferred language information in the user profile.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

String lang: Language information as defined by the ISO 3166 standard (FR, DE, ...).

#### **Returns**

Boolean: Possible values are "true" if the information was successfully changed, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.1.9 setScreenSize**

#### **Definition**

Sets the screen size of the device in the database.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

String `screenSize`: Screen size for the device - possible values:

- "maxi" if the device is a laptop,
- "mini" if the device is cell phone (palm, pocket pc ...).

#### **Returns**

Boolean: Possible values are "true" if the information was successfully changed, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

## **4.2 User profile information**

### **4.2.1 SetUserProfile**

#### **Definition**

Stores the device information in the database, in respect of an access key (unique for each client application) and the MAC address of the device.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

String `applicationName`: Access key for the client application calling this method.

String `applicationValues`: XML string to store.

#### **Returns**

Boolean: Possible values are "true" if the information was successfully stored, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.2.2 getUserProfile**

#### **Definition**

Returns the device information as it was stored in the database, in respect of an access key (unique for each client application) and the MAC address of the device.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

String `applicationName`: Access key for the client application calling this method.

#### **Returns**

String: XML string containing the device information stored in the database.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

## **4.3 WURFL capabilities**

### **4.3.1 canFlashLite10**

#### **Definition**

Checks whether the device can run Flash Lite 1.0.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can run Flash Lite 1.0, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.2 canFlashLite11**

#### **Definition**

Checks whether the device can run Flash Lite 1.1.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can run Flash Lite 1.1, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.3 canHtmlWeb32**

#### **Definition**

Checks whether the device can handle HTML 3.2.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can handle HTML 3.2, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### ***4.3.4 canHtmlWeb40***

##### **Definition**

Checks whether the device can handle HTML 4.0.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Boolean: Possible values are "true" if the device can handle HTML 4.0, "false" otherwise.

##### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### ***4.3.5 canVideo3GPP***

##### **Definition**

Checks whether the device can run videos in 3GPP format.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Boolean: Possible values are "true" if the device can run videos in 3GPP format, "false" otherwise.

##### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### ***4.3.6 canVideo3GPP2***

##### **Definition**

Checks whether the device can run videos in 3GPP2 format.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Boolean: Possible values are "true" if the device can run videos in 3GPP2 format, "false" otherwise.

##### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.7 *canVideoMOV***

#### **Definition**

Checks whether the device can run videos in MOV format.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can run videos in MOV format, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.8 *canVideoMP4***

#### **Definition**

Checks whether the device can run videos in MP4 format.

#### **Input parameters**

String `macAddress`: MAC address of the device – must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can run videos in MP4 format, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.9 *canVideoRealMedia10***

#### **Definition**

Checks whether the device can run videos in Real Media 10 format.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can run videos in Real Media 10 format, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### **4.3.10 *canVideoRealMedia8***

##### **Definition**

Checks whether the device can run videos in Real Media 8 format.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Boolean: Possible values are "true" if the device can run videos in Real Media 8 format, "false" otherwise.

##### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### **4.3.11 *canVideoRealMedia9***

##### **Definition**

Checks whether the device can run videos in Real Media 9 format.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Boolean: Possible values are "true" if the device can run videos in Real Media 9 format, "false" otherwise.

##### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### **4.3.12 *canVideoWMV***

##### **Definition**

Checks whether the device can run videos in WMV format.

##### **Input parameters**

String `macAddress`: MAC address of the device – must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Boolean: Possible values are "true" if the device can run videos in WMV format, "false" otherwise.

##### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.13 *canXhtmlBasic***

#### **Definition**

Checks whether the device supports the XHTML Basic document type.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device supports the XHTML Basic document type, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.14 *hasHttpsSupport***

#### **Definition**

Checks whether the device supports the HTTPS protocol.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device supports the HTTPS protocol, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.15 *hasPointingDevice***

#### **Definition**

Checks whether the device has a pointing device.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device has a pointing device, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### ***4.3.16 hasVideoSupport***

#### **Definition**

Checks whether the device can read videos.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Boolean: Possible values are "true" if the device can read videos, "false" otherwise.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### ***4.3.17 maxImageHeight***

#### **Definition**

Returns the maximum pictures height for the device.

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Int: The maximum pictures height (in pixel).

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### ***4.3.18 maxImagewidth***

#### **Definition**

Returns the maximum pictures width for the device

#### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

Int: The maximum pictures width (in pixel).

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.3.19 uaProf**

#### **Definition**

Returns the user agent of a device.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

#### **Returns**

String: The user agent for this device.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

## **4.4 Identification**

### **4.4.1 GetMAC**

#### **Definition**

Returns the MAC address of the device in respect of an IP address.

#### **Input parameters**

String ipAddress: IP address of the device.

#### **Returns**

String: The MAC address of the device.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

## **4.5 Users**

### **4.5.1 Authenticate**

#### **Definition**

Authenticates a user according to a user name, a password and an access key.

#### **Input parameters**

String userName: User name.

String password: Password.

String applicationKey: Access key for the client application calling this method.

#### **Returns**

Long: User's identification number.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.5.2 GetUserById**

#### **Definition**

Returns the user's information in respect of the user's identification number and an access key.

#### **Input parameters**

Long userId: User's identification number.

String applicationKey: Access key for the client application calling this method.

#### **Returns**

UserInfo: User's information.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.5.3 *getUserByMacAddress***

#### **Definition**

Returns the user's information in respect of a MAC address.

#### **Input parameters**

String *macAddress*: MAC address of the device - must respect the format *xx:xx:xx:xx:xx:xx*.

String *applicationKey*: Access key for the client application calling this method.

#### **Returns**

*UserInfo*: User's information.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

## **4.6 *Location***

### **4.6.1 *getApPosition***

#### **Definition**

Returns an Access Point's location in respect of the WGS84 reference.

#### **Input parameters**

Int *apId*: Access Point identification number.

#### **Returns**

*Position*: Access Point's location with latitude and longitude.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.6.2 *getApPositionXY***

#### **Definition**

Returns an Access Point's location in respect of the LUREF projection system.

#### **Input parameters**

Int *apId*: Access Point identification number.

#### **Returns**

*PositionXY*: Access Point's location with X and Y coordinates in respect of the LUREF projection system.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### **4.6.3 *getUserAp***

#### **Definition**

Returns the location of the Access Point on which the user is connected - the location is provided in respect of the WGS84 reference.

#### **Input parameters**

String *macAddress*: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

*UserAp*: Access Point's location with latitude and longitude.

#### **Exception**

If a runtime error occurs, an exception `"lu.pbs.hotcity.lbs.exception.LbsException"` is raised.

### **4.6.4 *getUserApXY***

#### **Definition**

Returns the location of the Access Point on which the user is connected - the location is provided in respect of the LUREF projection system.

#### **Input parameters**

String *macAddress*: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

#### **Returns**

*UserApXY*: Access Point's location with X and Y coordinates in respect of the LUREF projection system.

#### **Exception**

If a runtime error occurs, an exception `"lu.pbs.hotcity.lbs.exception.LbsException"` is raised.

#### ***4.6.5 getUserDistanceFromAp***

##### **Definition**

Returns the distance between the device and the Access Point on which the device is connected.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

Float: Distance between the device and the Access Point.

##### **Exception**

If a runtime error occurs, an exception `"lu.pbs.hotcity.lbs.exception.LbsException"` is raised.

#### ***4.6.6 getUserHistoryAp***

##### **Definition**

Returns the chronological list of connections for a device, during the last 48 hours, across all Access Points.

##### **Input parameters**

String `macAddress`: MAC address of the device - must respect the format `xx:xx:xx:xx:xx:xx`.

##### **Returns**

List<HistDist>: List of data structures `HistDist`.

##### **Exception**

If a runtime error occurs, an exception `"lu.pbs.hotcity.lbs.exception.LbsException"` is raised.

### ***4.6.7 getUserHistoryApFromDate***

#### **Definition**

Returns the chronological list of connections for a device, across all Access Points, since a given date.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

String fromDate: Date encoded as a String in the format « dd/MM/yyyy HH:mm:ss ».

#### **Returns**

List<HistDist>: List of data structures HistDist.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### ***4.6.8 getUserKnownAp***

#### **Definition**

Returns a list of Access Points, close to the user location.

#### **Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

#### **Returns**

List<UserAp>: List of data structures UserAp.

#### **Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### ***4.6.9 getUserPosition***

**Definition**

Returns a user's location in respect of the WGS84 reference.

**Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

**Returns**

Position: User's location with latitude and longitude in respect of the WGS84 reference.

**Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

#### ***4.6.10 getUserPositionXY***

**Definition**

Returns a user's location in respect of the LUREF projection system.

**Input parameters**

String macAddress: MAC address of the device - must respect the format xx:xx:xx:xx:xx:xx.

**Returns**

PositionXY: User's location with X and Y coordinates in respect of the LUREF projection system.

**Exception**

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

## 4.7 Maps

### 4.7.1 *getMap*

#### Definition

Returns a map corresponding to the location given in parameter, using the WGS84 reference.

#### Input parameters

String *applicationKey*: Access key for the client application calling this method.

Float *longitude*: Longitude in decimal.

Float *latitude*: Latitude in decimal.

String *scale*: Map size - possible values: "min" or "max".

Int *drawTarget*: If this parameter is not set to 0 (zero), draw a target icon to show the centered location.

#### Returns

String: URL to access the image.

#### Exception

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.

### 4.7.2 *getMapXY*

#### Definition

Returns a map corresponding to the location given in parameter, using the LUREF projection system.

#### Input parameters

String *applicationKey*: Access key for the client application calling this method.

Float *X*: Coordinate X in respect of the LUREF projection system.

Float *Y*: Coordinate Y in respect of the LUREF projection system.

String *scale*: Map size - possible values: "min" or "max".

Int *drawTarget*: If this parameter is not set to 0 (zero), draw a target icon to show the centered location.

#### Returns

String: URL to access the image.

#### Exception

If a runtime error occurs, an exception "lu.pbs.hotcity.lbs.exception.LbsException" is raised.