

类别 研发文档

编号 QXSK-D-WS0

版本 1.0

密级 公开

全息时空运营服务平台 WebSocket 接口

编制单位: 江苏凌比特微电子科技有限公司

编制人: 任珊虹

编制日期: 2018 年 03 月 27 日

修订记录

时间	作者	版本	说明
18年3月27日	任珊虹	1	初始版本

对于文档中的错误或者描述不清的地方，请联系作者，谢谢！

目 录

1 概述.....	5
1.1 关于全息时空北斗服务运营平台.....	5
1.2 平台提供的数据服务.....	5
1.3 本文档的内容.....	6
1.4 其它文档.....	6
2 WebSocket 的使用方式.....	7
2.1 一个 javascript 应用 WebSocket 的例子.....	7
2.2 一个 Android 应用 WebSocket 的例子.....	8
2.3 心跳机制.....	10
2.4 服务器监视 WebSocket 状态.....	10
3 平台的 WebSocket 接口.....	11
3.1 WebSocket 地址的格式.....	11
3.2 连接参数.....	11
3.3 数据参数.....	11
3.4 WebSocket 地址的清单.....	12
3.5 WebSocket 地址：terminalSocket.....	13
3.5.1 作用.....	13
3.5.2 地址示例.....	13
3.5.3 数据.....	13
3.6 WebSocket 地址：terminalLocationSocket.....	14
3.6.1 作用.....	14
3.6.2 地址示例.....	14
3.6.3 数据.....	14
3.7 WebSocket 地址：terminalStatusSocket.....	15
3.7.1 作用.....	15
3.7.2 地址示例.....	15
3.7.3 数据.....	15
3.8 WebSocket 地址：terminalStatusInfoSocket.....	16
3.8.1 作用.....	16
3.8.2 地址示例.....	16
3.8.3 数据.....	16
3.9 WebSocket 地址：terminalStatusNumberSocket.....	16
3.9.1 作用.....	16
3.9.2 地址示例.....	17
3.9.3 数据.....	17
3.10 WebSocket 地址：userBdMessageNewSocket.....	17
3.10.1 作用.....	17
3.10.2 地址示例.....	17
3.10.3 数据.....	17
3.11 WebSocket 地址：userBdMessageNewForMessageSocket.....	17
3.11.1 作用.....	17
3.11.2 地址示例.....	18
3.11.3 数据.....	18
3.12 WebSocket 地址：userBdMessageNewForNumberSocket.....	18
3.12.1 作用.....	18
3.12.2 地址示例.....	18
3.12.3 数据.....	18
3.13 WebSocket 地址：userBdMessageNewForTerminalSocket.....	18
3.13.1 作用.....	18

3.13.2 地址示例.....	18
3.13.3 数据.....	18
3.14 WebSocket 地址: terminalMapPointHtmlSocket.....	19
3.14.1 作用.....	19
3.14.2 地址示例.....	19
3.14.3 数据.....	19
3.15 WebSocket 地址: terminalMapPointStringSocket.....	19
3.15.1 作用.....	19
3.15.2 地址示例.....	19
3.15.3 数据.....	19

1 概述

1.1 关于全息时空北斗服务运营平台

全息时空北斗视野是全息时空北斗服务运营平台的客户端。平台对外服务的网址如下：
<http://www.qxsk9.com>

用户可以在此地址访问和下载平台的软件和文档。如下图：



资源	最新版本	版本时间	运行环境	访问/获得方式	文档
网页客户端 (服务平台功能)	v5.12	2018-02-28	支持HTML5的浏览器	进入 (建议Firefox和Safari) 进入(安全链接) (startssl证书, 已失效)	系统简介 《服务平台用户手册》 《运行环境的安装与配置手册-V3.3》 一些误读及其解释 北斗终端的接入方法 非北斗的数据终端的接入方法
安卓应用 (三合一) 客户端、本地定位、北斗伴侣	v4.0	2018-03-22	Android 4.0.2以上	 下载 或者扫描二维码: (在微信里扫描或点击均无效) (若用户已安装2.0以前的版本, 则须先卸载旧版再安装新版)	《用户手册》(待更新)
网络服务接口	v2.0	2016-05-07	RESTful(XML或JSON)		《网络服务协议-2.0》(待更新)
WebSocket编程接口	v1.0	2017-04-08	浏览器/iOS/Android		N/A
苹果客户端	v1.2	2016-12-16	iOS 8.0以上	“北斗视野” (目前在苹果商店已下架)	《用户手册》
北斗伴侣增强版 (北斗终端应用)	v1.3a	2016-11-05	安卓4.0.2以上 (配合E200使用)	 下载 或者扫描二维码: (在微信里扫描或点击均无效)	《用户手册》

©该网站版权归江苏凌比特微电子有限公司所有 备案/许可证号: 苏ICP备16007070号-1

1.2 平台提供的数据服务

全息时空北斗服务运营平台对外（客户端、第三方应用）提供以下两种数据服务：

RESTful Web Service: 由使用者向服务平台主动发出数据申请

WebSocket: 由服务平台主动推送数据给已连接的使用者
 即 WebSocket 是一种“订阅者”的数据接收模式。

Web Service 的适用场景是：服务使用者一次性获取数据即完成任务。

WebSocket 的适用场景是：服务使用者需要跟踪数据变化才能完成任务。

较常见的应用场景是：服务使用者利用 Web Service 获取一次性数据和初始化数据、然后利用 WebSocket 监听最新数据。

1.3 本文档的内容

本文档描述全息时空北斗服务运营平台提供的 WebSocket 接口。
文档主要包括以下内容：

- 1) 概述
- 2) WebSocket 的使用方式
- 3) 平台的 WebSocket 接口

本文档的下载地址为：

<http://www.qxsk9.com/QXSK-Platform-WebSocket.pdf>

1.4 其它文档

用户可以参考以下文档以了解更多的内容：

《全息时空北斗服务运营平台-用户手册》

<http://www.qxsk9.com/QXSK-Platform-UserGuide.pdf>

《全息时空北斗视野-安卓应用-用户手册》

<http://www.qxsk9.com/BeidouView-android.pdf>

《全息时空北斗视野-苹果客户端-用户手册》

<http://www.qxsk9.com/BeidouView-ios.pdf>

《全息时空北斗服务-网络服务协议》

<http://www.qxsk9.com/QXSK-Platform-WebService.pdf>

2 WebSocket 的使用方式

WebSocket 是近年来才实现标准化的数据交换模式。以往需要追踪数据变化的应用场景，只能采用客户端轮询或者专门的即时通讯平台，而 WebSocket 非常简洁地满足了服务器与客户端的双向通讯。

2.1 一个 javascript 应用 WebSocket 的例子

以下是一段 javascript 代码：网页打开时连接一个 WebSocket 服务，然后利用回调函数接收处理数据。

```
<script type = "text/javascript">
    var currentWebsocket;
    var websocketURL = "ws://www.qxsk9.com/QXSKweb/terminalSocket?user=x&wpass=y ";
    function connectWebSocket() {
        currentWebsocket = new WebSocket(websocketURL);
        currentWebsocket.onmessage = onMessageFromWebSocket;
        currentWebsocket.onopen = function () {
            currentWebsocket.send("Hello"); // 向服务器发送反馈或者参数
        };
        currentWebsocket.onclose = function () {
        };
        currentWebsocket.onerror = function () {
        };
    }
    function onMessageFromWebSocket(evt) {
        var data = parse(evt.data + "");
        // 处理接收到数据
    }
    function disconnectWebSocket() {
        if (currentWebsocket)
            currentWebsocket.close();
    }
    window.addEventListener("load", connectWebSocket(), false);
</script>
```

这个例子显示了应用 WebSocket 的主要框架：

- 1) 调用 `new WebSocket(地址)` 与服务器握手
- 2) 编写回调函数 `onopen`、`onmessage`、`onerror`、`onclose`

- 握手后随时向服务器发送数据、在回调函数中也可向服务器发送反馈。
- 在 `onmessage` 的回调函数中处理接收的数据。
- 握手后随时可以关闭连接、结束通信。

实际运行的代码还需要加入心跳（目前多数服务器或者浏览器并没有实现 ping/pong 机制）和异常处理。

以下是一个 JSF 应用 WebSocket 的截图：



2.2 一个 Android 应用 WebSocket 的例子

以下是一段 Android 代码：构建 `OkHttpClient`、然后定义 `WebSocketListener` 监听器来处理事件。

```
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import okhttp3.WebSocket;
import okhttp3.WebSocketListener;
import okio.ByteString;
```



```

public void open() {
    Request request = new Request.Builder().url(socketPath).build();
    OkHttpClient webSocketClient = new OkHttpClient.Builder()
        .retryOnConnectionFailure(true)
        .connectTimeout(8, TimeUnit.SECONDS)
        .readTimeout(5, TimeUnit.SECONDS)
        .writeTimeout(5, TimeUnit.SECONDS)
        .build();
    webSocketClient.newWebSocket(request, new AppWebSocketListener());
    webSocketClient.dispatcher().executorService().shutdown();
}

private final class AppWebSocketListener extends WebSocketListener {
    @Override
    public void onOpen(WebSocket webSocket, Response response) {
        webSocket.send("Hello"); // 向服务器发送反馈或者参数
    }
    @Override
    public void onMessage(WebSocket webSocket, String text) {
        // 处理接收到的字符串数据 text
    }
    @Override
    public void onMessage(WebSocket webSocket, ByteString bytes) {
        // 处理接收到的字节流数据 bytes
    }
    @Override
    public void onClosing(WebSocket webSocket, int code, String reason) {
        Log.d("WebSocket", "onClosing: " + code + "/" + reason);
    }
    @Override
    public void onClosed(WebSocket webSocket, int code, String reason) {
        Log.d("WebSocket", "onClosed: " + code + "/" + reason);
    }
    @Override
    public void onFailure(WebSocket webSocket, Throwable t, Response response) {
        webSocket.close(1000, null);
    }
}
}

```

2.3 心跳机制

WebSocket 协议包含 ping/pong 机制，来保证通信不会因为一段时间没数据交换而被底层网络协议强行关闭。但是目前至少 GlassFish 和 FireFox 并没有协调起来实现这种心跳。

因此，无论是浏览器客户端的代码、还是智能终端的代码，都需要自行实现 WebSocket 的心跳：在服务器和客户端之间定时发生一次数据传输。根据实践，5 分钟是一个合适的间隔。

另外，浏览器客户端还需要注意实现 Http 心跳，以免页面长时间无用户响应而被服务器强行中断 session。例如，一个页面利用 WebSocket 持续监视数据变化，而长时间没有用户输入，此时就需要浏览器定时向服务器发送数据来延续 http 的 session。这个间隔可以是 session 寿命减去几秒即可。

2.4 服务器监视 WebSocket 状态

以下代码示例 WebSocket 关闭原因：

```
public static String getCloseReason(int closeCodes) {  
    switch (closeCodes) {  
        case 1000:  
            return "正常关闭";  
        case 1001:  
            return "离开（going away），例如服务器关闭或浏览器导航到其他页面";  
        case 1002:  
            return "端点因为协议错误而终止连接";  
        case 1003:  
            return "端点由于它收到了不能接收的数据类型而终止连接";  
        case 1007:  
            return "端点因为消息中接收到的数据是不符合消息类型而终止连接";  
        case 1008:  
            return "端点因为接收到的消息违反其策略而终止连接";  
        case 1009:  
            return "端点因接收到的消息对它的处理来说太大而终止连接";  
        case 1010:  
            return "端点（客户端）因为它期望服务器协商一个或多个扩展，但服务器没有在 WebSocket 握手响应消息中返回它们而终止连接";  
        case 1011:  
            return "服务器端因为遇到了一个不期望的情况使它无法满足请求而终止连接。";  
        default:  
            return "原因未知";  
    }  
}
```

3 平台的 WebSocket 接口

3.1 WebSocket 地址的格式

目前平台对外提供 11 个 WebSocket 地址，格式是：

`ws://服务器地址/WebSocket 地址?连接参数`

3.2 连接参数

连接参数包含账户信息，可以是以下三种格式之一：

`ws://www.qxsk9.com/QXSKweb/WebSocket 地址?user=用户名&wpass=以种子 1 进行 PBE 加密后的密码`

`ws://www.qxsk9.com/QXSKweb/WebSocket 地址?user=用户名&apass=以种子 2 进行 PBE 加密后的密码`

`ws://www.qxsk9.com/QXSKweb/WebSocket 地址?user=用户名&pass=未加密的密码`

账户参数是必须的。若账户信息非法，则 WebSocket 连接失败。

客户端若要用加密的密码，则种子、盐、和加密算法都需要与服务器一致。（算法只对合作方开放）

连接参数还可以包括客户端类型：

“`ctype=Web_Client`”表示客户端为浏览器

“`ctype=A`”表示客户端为安卓终端

“`ctype=I`”表示客户端为 iOS 终端

连接参数还可以包括客户端地址，例如：

`cid=49.80.112.33 江苏省`

参数 `ctype` 和 `cid` 用于系统日志，可以不提供。

3.3 数据参数

建立 WebSocket 连接后，客户端需要向服务平台发送以下参数以后才可以接收到数据：

"MonitoringTerminals"：需要监视的终端名列表，以逗号分隔。若要监视自己权限内的所有终端，则把这个参数设为 "*" 或者空值。

"SessionLanguage"：客户端的语言，缺省是 "zh_CN"。服务器的数据是国际化了的，可按照客户端要求推送本地化了的数据。

客户端可以把参数打包成 Json 数据发送给服务器。

以下是 Javascript 示例代码：在连接打开后，立即向服务器发送数据参数，然后等待数据。

```
currentWebsocket.onopen = function () {
    var clientAttributes = {
        "MonitoringTerminals": "001-模拟终端-GPS",
        "SessionLanguage": "zh_CN"
    };
    currentWebsocket.send(JSON.stringify(clientAttributes));
}
```

以下是 Android 代码示例：连接打开后，数据参数以 Json 字串发送给服务器。

```
private class TerminalLocationListener extends WebSocketListener {
    @Override
    public void onOpen(WebSocket websocket, Response response) {
        JSONObject clientAttributes = new JSONObject();
        try {
            clientAttributes.put("SessionLanguage", "zh_CN");
            clientAttributes.put("MonitoringTerminals","*");
            websocket.send(clientAttributes.toString());
        } catch (JSONException e) {
            websocket.close(1003, "初始化参数错误");
        }
    }
}
```

3.4 WebSocket 地址的清单

地址	作用	数据格式
terminalSocket	终端属性发生变化时，客户端收到终端的当前的全部属性	Json
terminalLocationSocket	终端有新位置时，客户端收到新位置的全部属性	Json
terminalStatusSocket	终端状态发生增/删/改/清除，客户端收到该状态的全部属性	Json
terminalStatusInfoSocket	终端状态发生增/删/改/清除，客户端收到该状态的改变特征	字符串
terminalStatusNumberSocket	终端状态发生增/删/清除，客户端收到该状态的改变特征	字符
userBdMessageNewSocket	终端有新用户北斗消息时，客户端收到该新消息的全部属性	Json
userBdMessageNewForMessageSocket	终端有新用户北斗消息时，客户端收到该新消息的内容及时间	Json
userBdMessageNewForNumberSocket	终端有新用户北斗消息时，客户端收到一个字符	字符“n”
userBdMessageNewForTerminalSocket	终端有新用户北斗消息时，客户端收到相应的终端名	字符串
terminalMapPointHtmlSocket	终端属性变化时，客户端收到在 html 地图上更新的数据	Json
terminalMapPointStringSocket	终端属性变化时，客户端收到智能终端地图上更新的数据	Json

3.5 WebSocket 地址: terminalSocket

3.5.1 作用

只要终端属性发生变化，则客户端就会收到终端的当前的全部属性。

3.5.2 地址示例

ws://www.qxsk9.com/QXSKweb/terminalSocket?user=Mara&wpass=5BF07EC1C4E1419D2E&ctype=Web_Client

3.5.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 `getTime()` 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```
{
  "terminalid":60,
  "register_number":"001-模拟终端-GPS",
  "last_x":118.67991,
  "last_y":32.16652,
  "last_z":23,
  "last_speed":100,
  "last_direction":4,
  "last_location_time":1522119443755,
  "last_status":
  {
    "dataid":221,
    "terminal":"001-模拟终端-GPS",
    "status_type":"OutFenceAlert",
    "record_x":118.68061,
    "record_y":32.16812,
    "record_z":23,
    "record_time":1510897109000,
    "fence":"z",
    "alert_message":"hello",
    "email_sent":false,
    "message_sent":false,
    "feedback_sent":false,
    "cleared":false,
    "cleared_time":1522119433629
    "repeats":84825,
    "last_record_time":1522119433629
  }
}
```

```

"bracelet_data":
  {
    "dataid":30,
    "terminal":"001-模拟终端-GPS",
    "x":118.68061,
    "y":32.16812,
    "steps":835,
    "calorie":36,
    "sleep":0.0,
    "pulse":0,
    "body_temperature":26.1,
    "env_temperature":26,
    "humidity":37,
    "pressure":1008.5,
    "record_time":1522119433620
  },
  "last_status_time":1522119433629
}

```

3.6 WebSocket 地址: terminalLocationSocket

3.6.1 作用

只要终端有了新位置数据，则客户端就会收到该新位置数据的全部属性。

3.6.2 地址示例

<ws://www.qxsk9.com/QXSKweb/terminalLocationSocket?user=任珊虹&wpass=5BF07EC1C4E1419D2E0>

3.6.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 `getTime()` 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```

{
  "terminal":"001-模拟终端-GPS",
  "x":118.69611,
  "y":32.15522,
  "z":67,
  "speed":185,
  "direction":6,
  "record_time":1522120973455
}

```

```
}
```

3.7 WebSocket 地址: terminalStatusSocket

3.7.1 作用

只要终端状态发生任何变化（增/删/改/清除），则客户端就会收到该状态的全部属性。

3.7.2 地址示例

<ws://www.qxsk9.com/QXSKweb/terminalStatusSocket?user=任珊虹&wpass=5BF07EC1C4E1419D2E0>

3.7.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 `getTime()` 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```
{
  "dataid":221,
  "terminal":"001-模拟终端-GPS",
  "status_type":"OutFenceAlert",
  "record_x":118.68061,
  "record_y":32.16812,
  "record_z":23,
  "record_time":1510897109000,
  "fence":"z",
  "alert_message":"hello",
  "email_sent":false,
  "message_sent":false,
  "feedback_sent":false,
  "cleared":false,
  "cleared_time":1522119433629
  "repeats":84825,
  "last_record_time":1522119433629
  "bracelet_data":
  {
    "dataid":30,
    "terminal":"001-模拟终端-GPS",
    "x":118.68061,
    "y":32.16812,
    "steps":835,
    "calorie":36,
    "sleep":0.0,
```

```

        "pulse":0,
        "body_temperature":26.1,
        "env_temperature":26,
        "humidity":37,
        "pressure":1008.5,
        "record_time":1522119433620
    },
    "updateType":"cleared"
}

```

特别的，“updateType”字段可能的取值为：“new”、“updated”、“cleared”、“deleted”，对应于终端状态的四种变化。

3.8 WebSocket 地址：terminalStatusInfoSocket

3.8.1 作用

只要终端状态发生任何变化（增/删/改/清除），则客户端就会收到该状态的改变特征。

3.8.2 地址示例

<ws://www.qxsk9.com/QXSKweb/terminalStatusInfoSocket?user=任珊虹&wpass=5BF07EC1C4E1419D2E0>

3.8.3 数据

服务器推送的是字符串，格式为：

特征字符:终端名 状态类型 围栏名（如果有）

其中，“特征字符”定义为：

“n”表示新状态

“u”表示状态更新

“c”表示状态被清除

“d”表示状态被删除

状态类型可能的取值为：出围栏通知、进围栏通知、出围栏警报、进围栏警报、SOS 警报。状态类型的取值与客户端发送的语言参数有关，即若客户端发送的语言参数是英文，则取值为英文。

以下是一个数据示例：

u:001-模拟终端-GPS 出围栏警报 离开珍珠泉区域

3.9 WebSocket 地址：terminalStatusNumberSocket

3.9.1 作用

只要终端状态发生数量变化（增/删/清除），则客户端就会收到该状态的改变特征。

3.9.2 地址示例

<ws://www.qxsk9.com/QXSKweb/terminalStatusNumberSocket?user=任珊虹&wpass=5BF07EC1C4E1419D2E0>

3.9.3 数据

服务器推送的是一个字符：

“n”表示新状态

“c”表示状态被清除

“d”表示状态被删除

3.10 WebSocket 地址：userBdMessageNewSocket

3.10.1 作用

只要终端有了新的用户北斗消息，则客户端就会收到该新消息的全部属性。

3.10.2 地址示例

<ws://www.qxsk9.com/QXSKweb/userBdMessageNewSocket?user=任珊虹&wpass=5BF07EC1C4E1419D2E0>

3.10.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 getTime() 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```
{
  "dataid":10656,
  "terminal":"000-模拟终端-消息",
  "from_terminal":true,
  "message":"羌笛何须怨杨柳，春光不度玉门关\n《凉州词二首》 唐代·王之涣",
  "user":"试用账户",
  "phone":"111111111",
  "record_time":1522126338000,
  "is_urgent":false,
  "is_success":false
  "sent_time":1522126338000,
}
```

3.11 WebSocket 地址：userBdMessageNewForMessageSocket

3.11.1 作用

只要终端有了新的用户北斗消息，则客户端就会收到该新消息的内容及其时间。

3.11.2 地址示例

<ws://www.qxsk9.com/QXSKweb/userBdMessageNewForMessageSocket?user=任珊虹&wpass=5BF07EC1C4E1419>

3.11.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 `getTime()` 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```
{
  "terminal": "000-模拟终端-消息",
  "message": "江深竹静两三家，多事红花映白花\n《江畔独步寻花七绝句》 唐代·杜甫",
  "record_time": 1522126564000
}
```

3.12 WebSocket 地址：userBdMessageNewForNumberSocket

3.12.1 作用

只要终端有了新的用户北斗消息，则客户端就会收到一个字符。

3.12.2 地址示例

<ws://www.qxsk9.com/QXSKweb/userBdMessageNewForNumberSocket?user=任珊虹&wpass=5BF07EC1C4E1419>

3.12.3 数据

服务器推送的是一个字符“n”，表示有新消息。

3.13 WebSocket 地址：userBdMessageNewForTerminalSocket

3.13.1 作用

只要终端有了新的用户北斗消息，则客户端就会收到相应的终端名。

3.13.2 地址示例

<ws://www.qxsk9.com/QXSKweb/userBdMessageNewForTerminalSocket?user=任珊虹&wpass=5BF07EC1C4E1419>

3.13.3 数据

服务器推送的是一个字符串，表示有收到新消息的终端名。

3.14 WebSocket 地址: terminalMapPointHtmlSocket

3.14.1 作用

只要终端的属性发生变化，则客户端就会收到终端在 html 地图上需要更新的数据。

3.14.2 地址示例

<ws://www.qxsk9.com/QXSKweb/terminalMapPointHtmlSocket?user=任珊虹&wpass=5BF07EC1C4E1419>

3.14.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 getTime() 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```
{
  "name": "001-模拟终端-GPS",
  "x": 118.692573,
  "y": 32.172227,
  "popInfo": "<tr><td>终端注册号:</td><td>001-模拟终端-GPS</td></tr><tr><td>SIM卡号:</td><td></td></tr><tr><td>北斗卡号:</td><td></td></tr><tr><td>最后位置:</td><td>118.68101,32.16802</td></tr><tr><td>最后的速度:</td><td></td></tr><tr><td>最后方向:</td><td></td></tr><tr><td>最后位置的记录时间:</td><td>2018-03-27 13:03:43</td></tr><tr><td>最后的状态:</td><td>出围栏警报 - 离开珍珠泉区域</td></tr><tr><td>最后状态的记录时间:</td><td>2018-03-27 13:03:43</td></tr><tr><td>最新消息:</td><td></td></tr><tr><td>最新消息的时间:</td><td></td></tr><tr><td>照片:</td><td><a href='/QXSKweb/resources/images/terminalPhotos/60.jpg' target='_blank'><img src='/QXSKweb/resources/images/terminalPhotos/60.jpg' height=100px</a></td></tr>",
  "status": "OutFenceAlert",
  "time": 1522127023483
}
```

特别的，“popInfo”是已经格式化好了的、该终端在地图上弹出信息的 html 代码。

3.15 WebSocket 地址: terminalMapPointStringSocket

3.15.1 作用

只要终端的属性发生变化，则客户端就会收到终端在智能终端地图上需要更新的数据。

3.15.2 地址示例

<ws://www.qxsk9.com/QXSKweb/terminalMapPointStringSocket?user=任珊虹&wpass=5BF07EC1C4E1419>

3.15.3 数据

服务器推送的是 Json 数据，若某个字段无值，则不出现在数据里。时间值均为 getTime() 得到的 long 型。

以下是一个数据示例，包含本地址推送数据的所有字段：

```
{  
  "name": "001-模拟终端-GPS",  
  "x": 118.688451,  
  "y": 32.164635,  
  "popInfo": "终端注册号:001-模拟终端-GPS\nSIM 卡号:\n北斗卡号:\n最后位置:118.67691, 32.16042\n最后的速度:\n最后方向:\n最后位置的记录时间:2018-03-27 13:11:03\n最后的状态:出围栏警报 - 离开珍珠泉区域\n最后状态的记录时间:2018-03-27 13:11:03\n最新消息:\n最新消息的时间:\n",  
  "status": "OutFenceAlert",  
  "time": 1522127463532  
}
```

特别的，“popInfo”是已经格式化好了的、该终端在智能终端地图上弹出信息的字符串。

<文档结束>