目录

[1.  概述 1](#_Toc425797012)

[1.1 目的 1](#_Toc425797013)

[1.2 适用范围 1](#_Toc425797014)

[1.3请求方式 1](#_Toc425797015)

[1.4通用参数 2](#_Toc425797016)

[2.  数据结构 2](#_Toc425797017)

[2.1语法规则 2](#_Toc425797018)

[3.  设备操作接口定义 3](#_Toc425797019)

[3.1授权方式 3](#_Toc425797020)

[3.2接口列表 4](#_Toc425797021)

[3.3接口URL 5](#_Toc425797022)

[4.  附录 12](#_Toc425797023)

[4.1 参数列表 12](#_Toc425797024)

## 1.  概述

### 1.1 目的

规范前后台交互的协作方式，保持前后台交互的一致性和项目开发技术及习惯的延续性。

### 1.2 适用范围

适用于公车管理通信系统开发人员。

### 1.3请求方式

API请求均采用POST或GET中的一种作为请求方式。对于用户敏感信息的传递，均使用POST请求方式，GET请求均应兼容POST请求。请求参数根据实际情况选择key/value对或json数据结构。

除用户权限认证接口外，应当以资源名称作为接口URL，资源名称均以复数形式表示。对该资源的不同操作，以cmd参数区分。

URL：

http://url-to-resource/{resource\_id}?{query\_string}

说明：

* URL里的resource为操作的资源名，这里的resource可能为channel固定值，也有可能是具体的channel\_id。
* {query\_string}由通用参数部分和具体API调用参数部分组成。
* {query\_string}中的key/value对都必须经过urlencode处理，且必须是UTF-8编码。
* 对于GET请求，{query\_string}必须放在QUERY参数中传递，即放在“?”后面。
* 对于POST请求，{query\_string}放在POST参数中传递，该参数可以为json或key/value对。
* GET参数请进行URL 编码，编码时请遵守 RFC 1738

### 1.4通用参数

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 是否必需 | 描述 |
| cmd | string | 否 | API的资源操作方法名。 |

## 2.  数据结构

前后台交互消息的数据结构是由控制属性和业务数据对象组成，统一使用JSON格式。所有交互消息要符合该结构规范。图2-1展示了数据结构的定义。清单2-1则举例说明生成的JSON样式。

### 2.1语法规则

#### 2.1.1JSON 名称/值对

JSON 数据的书写格式是：名称/值对,名称/值对包括字段名称（在双引号中），后面写一个冒号，然后是值：

"firstName" : "John"

#### 2.1.2 JSON 对象

JSON 对象在花括号中书写：对象可以包含多个名称/值对：

{ "firstName":"John" , "lastName":"Doe" }

#### 2.1.3JSON 数组

JSON 数组在方括号中书写：数组可包含多个对象：

{

"employees":

[

{ "firstName":"John" , "lastName":"Doe" },

{ "firstName":"Anna" , "lastName":"Smith" },

{ "firstName":"Peter" , "lastName":"Jones" }

]

}

## 3.  设备操作接口定义

### 3.1授权方式

#### 3.1.1授权信息格式

用户名和密码以 \t 分隔连接成字符串形式。

格式：{user} \t {password}

#### 3.1.2加密算法

使用AES算法对授权信息进行加密，密钥长度256bit，用BASE64进行二进制转码。

#### 3.1.3授权请求方式

将授权信息密文写入到HttpRequestHeader的Authorization标签请求Webapi方式。

#### 3.1.4授权密钥配置

将密码信息配置到web服务器的配置文件中格式如下

<key="D6B5A3F9F1F0A2D3S6D237A1D6B5A3F9F1F0A2D3S6D237A1D6B5A3F9F1F0A2D3S" type="aes" />

### 3.2接口列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 请求URL | HTTP方式 | 参数格式 | 返回格式 | 命令描述 |
| device/settings | GET | Key/value | Json | 查询终端参数指令 |
| device/settings | POST | Json | Json | 设置终端参数指令 |
| device/settings | POST | Json | Json | 恢复出厂设置 |
| device/gps | GET | Key/value | Json | 获取设备的GPS信息 |
| device/errors | POST | Json | Json | 清除故障信息 |
| device/message | POST | Json | Json | 文字信息发送 |

### 3.3接口URL

1. **查询终端参数**

**接口URL**

URL：

http://url-to-resource/device/settings?{query\_string}

**HTTP方式**:

GET

**输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 描述 | 例子 |
| cmd | String | 操作类型 | 该操作为“get” |
| id | String | 设备标识 | 10010 |
| tags | String | 标识ID列表  以16进制数值的字符串形式表示每个tagid,  用逗号分隔  **参考附录4.1.1** | F0F1,F2F3,F4F5,F6F7 |

**返回参数**

正确返回示例

Content-type: text/html; charset=utf-8

{

"ret":”0”,

“data”:{

"total":"12",

"fail\_tags":[{“tag”:”08A1”},{“tag”:”08A2”}],

"success\_tags":[

{“tag”:”08A1”,”values”:[{“val”:”a1”},{”val”:”a2”},{”val”:”a3”}] },

{“tag”:”08A2”,”values”:[{“val”:”a1”},{”val”:”a2”},{”val”:”a3”}] }

]

}

}

错误返回示例

Content-type: text/html; charset=utf-8

{

"ret":1002,

"msg":"认证错误"

}

1. **设置终端参数**

**接口URL**

URL：

http://url-to-resource/device/settings

**HTTP方式**

POST

**输入参数（flags:参考附录4.1.1）**

Content-type: text/html; charset=utf-8

{

“cmd”:”该操作为set”,

“id”: “10010”,

“flags”:[

{“tag”:”08A1”,”values”:[{“val”:”a1”},{”val”:”a2”},{”val”:”a3”}] },

{“tag”:”08A2”,”values”:[{“val”:”a1”},{”val”:”a2”},{”val”:”a3”}] }

]

}

**返回参数**

正确返回示例

Content-type: text/html; charset=utf-8

{

"ret":”0”,

"data":[

{"tag":"08A1"},

{"tag":"08A3"}

]

}

错误返回示例

Content-type: text/html; charset=utf-8

{

"ret":1002,

"msg":"认证错误"

}

1. **恢复出厂设置**

**接口URL**

URL：

http://url-to-resource/device/settings

**HTTP方式**

POST

**输入参数**

Content-type: text/html; charset=utf-8

{

“cmd”:”该操作为reset”,

“id”: “10010”

}

**返回参数**

正确返回示例

Content-type: text/html; charset=utf-8

{

"ret":”0”,

}

错误返回示例

Content-type: text/html; charset=utf-8

{

"ret":1002,

"msg":"认证错误"

}

1. **获取设备GPS信息**

**接口URL**

URL：

http://url-to-resource/device/gps?{query\_string}

**HTTP方式**

GET

**输入参数**

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名称 | 类型 | 描述 | 例子 |
| cmd | String | 命令类型 | 该操作为”get” |
| id | String | 设备标识 | 10010 |

**返回结果**

正确返回示例

Content-type: text/html; charset=utf-8

{

"ret":”0”,

“data”: {

“stat\_data”:{

“last\_accon\_time”:”29387934”,

”utc\_time”:”2938798937”,

“total\_trip\_mile”:”10”,

“current\_trip\_mile”:”10”,

“total\_fuel”:”10”,

“current\_fuel”:”10”,

“vstate”:”123456”,

“reserved”:[

{“val”:”0”},{“val”:”0”},{“val”:”0”},{“val”:”0”},

{“val”:”0”},{“val”:”0”},{“val”:”0”},{“val”:”0”}

]

},

“gps\_data”:[

{

“timestamp”:”2938798937”,”lat”:”12.20”,

”lon”:”87.29”,”speed”:”19.10”,”dir”:”1”,”flag”:”1”

},

{

“timestamp”:”2938798937”,”lat”:”12.20”,

”lon”:”87.29”,”speed”:”19.10”,”dir”:”1”,”flag”:”1”

}

]

}

}

错误返回示例

Content-type: text/html; charset=utf-8

{

"ret":1002,

"msg":"认证错误"

}

1. **清除故障信息**

**接口URL**

URL：

http://url-to-resource/device/errors

**HTTP方式**

POST

**输入参数**

Content-type: text/html; charset=utf-8

{

“cmd”:”该操作为clear”,

“id”: “10010”

}

**返回结果**

正确返回示例

Content-type: text/html; charset=utf-8

{

"ret":”0”,

}

错误返回示例

Content-type: text/html; charset=utf-8

{

"ret":1002,

"msg":"认证错误"

}

1. **发送文字信息**

**接口URL**

URL：

http://url-to-resource/device/message

**HTTP方式**

POST

**输入参数**

{

“cmd”:”send”,

“id”: “10010”,

“data”:”test messages”

}

**返回结果**

正确返回示例

Content-type: text/html; charset=utf-8

{

"ret":”0”,

}

错误返回示例

Content-type: text/html; charset=utf-8

{

"ret":1002,

"msg":"认证错误"

}

## 4.  附录

### 4.1 参数列表

参考文档《NEW OBD SMART通讯协议(CB212-C1005 Rev.4.26)（加密）.pdf》的“4.3 TLV描述列表”节

例如：超速告警(0x1001)，WebApi调用tag为 “1001”