上海艾拉比智能科技有限公司	文档编号		版本	V4.0	密级	商密 A
	项目名称					
	项目来源					

OTA 平台接口交互文档

`(内部资料请勿外传)

编写：	云平台部	日期：	2018/11/06
检查：		日期：	
审核：		日期：	
批准：		日期：	

上海艾拉比智能科技有限公司

文档变更记录

序号	变更（+/-）说明	作者	版本号	日期	批准
1	第一版编写		V3.0	2018/02/26	
2	第二版编写		V4.0	2018/11/06	
3	第三版编写		V4.1	2018/07/19	
4	第四版编写		V4.1.5	2019/10/18	
5	第五版编写		V4.1.6	2019/12/23	

版本变更记录

V4.1.5 版本变更

1. 增加注册接口和获取服务器配置接口时序图。
2. 修改接口文档格式规范。
3. 更新目录和报文信息。

V4.1.6 版本变更

1. Check 接口和 Report Vehicle Config 接口请求参数改为一致
2. GetServerConfig 接口不再下发 ecuPartNum
3. Check 接口下发车辆升级前置条件更改，可通过管理平台动态配置，后续车辆升级前置条件请以后台配置为准
4. 增加根证书下发接口

目　录

引　言

背景

随着汽车的越来越智能化，电子模块和软件技术在整车上的应用程度也越来越高，未来汽车电子软件问题也会愈加爆发，需要返修的可能性也会更大。OTA 技术的推广使用也会成为趋势。

（1）从智能角度来说，随着智能化的不断的普及和不断演进，需要通过 OTA 去更新算法和具体的程序的性能；

（2）从安全的角度来说，需要通过 OTA 去弥补信息安全里面的漏洞。OTA 跟信息安全是一个密不可分的整体。一方面，OTA 需要汽车信息安全体系的保驾护航，另一方面 OTA 也是解决汽车信息安全风险和漏洞的最佳工具；

（3）从用户体验的角度来说，更需要通过 OTA 来不断优化用户的体验，根据用户群体，创造差异化的驾乘体验，提升用户满意度。

编写目的和范围

编写本文档的主要目的是为了服务端研发人员、车端研发人员和测试人员提供工作依据和指南。研发和测试人员根据此文档进行相关代码开发和功能测试，为软件开发和维护提供基础。

本文档详细定义了车端与 OTA 平台的交互接口，包括状态码约定、加密和签名约定、接口说明、车端与服务端交互时序图、接口地址、请求参数、相应参数、字段长度、请求示例和响应示例等。

1 相关约定

1.1 编码方式

统一采用 UTF-8 编码方式。

1.2 上下行约定

1. 车辆向 OTA 平台发起请求称为上行。
2. OTA 平台向车辆推送信息称为下行。
3. 约定同一次检测、下载结果、升级结果等结果上报， 同一个 sessionId 表示。

1.3 交互方式约定

请求-响应协议：https、http
替代域名表示：{ota-server}

1.4 响应时间约定

查询相应时间：≤ 500 ms

1.5 错误码约定

1.5.1 服务端下发参数状态码定义

1000	成功
2001	暂无最新版本
2002	车辆未注册
2003	车辆不存在
3001	配置信息需要更新
3002	配置信息不存在
5001	参数 Json 格式转换异常
5002	参数校验失败
5003	签名验证失败
5004	文件保存异常
5005	请求报文解密失败
6000	任务中止

7001	暂无可下发根证书
10000	服务器内部异常
10001	禁止短时间内重复访问服务器

1.5.2 客户端状态码定义

见《车辆状态及错误码与事件汇总.xlsx》

SVN 地址:svn://svnhome.adfuture.cn:8888/fota/iot/vehicle_doc/爱驰 MAS861-U5 项目/05_详细设计/汽车端/OTA4.X 车辆状态及错误码与事件汇总

1.6 报文加密约定

1.6.1 报文加密说明

除升级日志文件上传和诊断日志文件上传接口请求报文暂不加密外，其余所有接口请求报文均使用加密，其中注册接口加密所使用的 key 为约定值，检测、下载上报、升级上报、事件上报接口加密报文所使用的 key 为车辆 secret 值。

所有接口的响应报文均使用加密，其中注册接口加密所使用的 key 为约定值，检测、下载上报、升级上报、事件上报、诊断日志文件上传、升级日志文件上传接口加密报文所使用的 key 为车辆 secret 值。

服务端可能存在尚未获取到车辆 secret 之前，就返回结果报文给车端的情况（如服务端获取 secret 失败，抛出异常），此时服务端返回的报文加密所使用的 key 为约定值。

1.6.2 报文加密算法

加密算法:AES-256，加密模式:ECB，填充:PKCS7Padding，key: 32 位字符，和车端约定，Base64 编码。

1.7 接口签名约定

1.7.1 签名生成规则

为了保证交互报文的安全性，每个接口请求/响应报文都需要全字段验签。

签名生成步骤：

第一步，设所有发送或者接收到的参数节点下的数据为集合 M，将集合 M 内非空参数值的参数（除了 sign 参数外）按照参数名 ASCII 码从小到大排序（字典序），使用 URL 键值对的格式（即 key1=value1&key2=value2…）拼接成字符串 stringA。StringA 就是签名内容。

特别注意以下重要规则：

◆参数名 ASCII 码从小到大排序（字典序）；
◆如果参数的值为空或空字符串不参与签名；
◆参数名区分大小写；
◆接口可能增加字段，验证签名时必须支持增加的扩展字段；
◆文件不参与签名；
◆除了注册接口，其他所有接口请求报文签名时都要增加 deviceId 字段；
◆字段 sign 不参与签名。

第二步，对 stringA 使用 key 进行 hmacmd5 算法签名，得到 sign。

以上报车辆信息接口为例，说明 stringA 生产过程：

请求参数（简化版）：

```
{
    “timestamp”:1529724024,
    “sign”:”2729c9e1e36dfee2f08c3487ed402dc1”,
    “networkInfo”: {
        “netType”: “wifi”,
        “lac”: “0210”,
        “cid”: “”
    },
    “ecus”:[
        {
            “ecuPartNum”:”ecu1”,
            “ecuSwid”:””,
            “ecuSVer”:”v1”,
        },
        {
            “ecuPartNum”:”ecu2”,
            “ecuSwid”:”3333”,
            “productDate”:”2018-05-22”,
        }
    ]
}
```

找到有效签名字段：

```
deviceId=a9f57b0e7dcc2cfde0b8ded025b2750b
timestamp=1529724024
networkInfo.netType=wifi
networkInfo.lac=0210
ecus[0].ecuPartNum=ecu1
ecus[0].ecuSVer=v1
ecus[1].ecuPartNum=ecu2
ecus[1].ecuSwid=3333
ecus[1].productDate=2018-05-22
```

排序并生产 stringA 内容：

deviceId=a9f57b0e7dcc2cfde0b8ded025b2750b&ecus[0].ecuPartNum=ecu1&ecus[0].ecuSVer=v1&ecus[1].ecuPartNum=ecu2&ecus[1].productDate=2018-05-22&ecus[1].ecuSwid=3333&networkInfo.netType=wifi&networkInfo.lac=0210&timestamp=1529724024

1.7.2 客户端请求报文加密及签名总结

接口	签名/加密	key
Register	签名	vin
	加密	约定 key
Get Server Config	签名	secret
	加密	secret
Report Vehicle Config	签名	secret
	加密	secret
Check	签名	secret
	加密	secret
Vehicle Result Report	签名	secret
	加密	secret
Download Speed Report	签名	secret
	加密	secret
File Upload	签名	secret
	暂不加密	无
Event Report	签名	secret
	加密	secret

1.7.3 服务端返回报文加密及签名总结

针对服务端响应报文，各接口全字段验签和报文加密使用的 Key 总结如下表:

接口	签名/加密	Http 响应状态码	
		200	**500**
Register	签名	vin	约定 key
	加密	约定 key	约定 key
Get Server Config	签名	secret	约定 key
	加密	secret	约定 key
Report Vehicle Config	签名	secret	约定 key
	加密	secret	约定 key
Check	签名	secret	约定 key

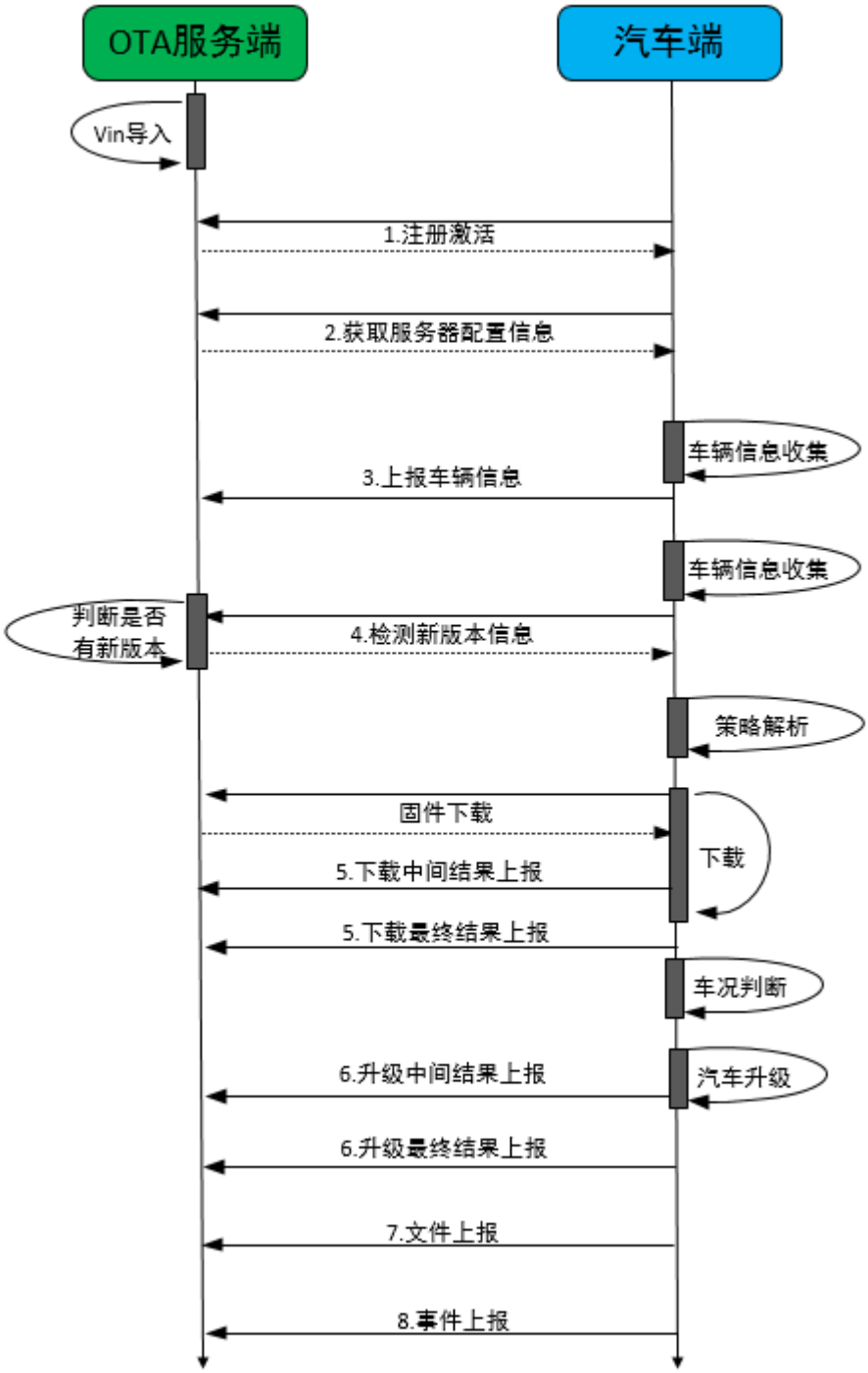
	加密	secret	约定 key
Vehicle Result Report	签名	secret	约定 key
	加密	secret	约定 key
Download Speed Report	签名	secret	约定 key
	加密	secret	约定 key
File Upload	签名	secret	约定 key
	加密	secret	约定 key
Event Report	签名	secret	约定 key
	加密	secret	约定 key

如注册接口若服务端返回 Http 响应状态码为 200，则车端应使用约定 key 解密报文，然后用 vin 验签；

如注册接口若服务端返回 Http 响应状态码为 500，则车端应使用约定 key 解密报文，然后用约定 key 验签；

2 接口设计

2.1 时序图


OTA服务端
汽车端
Vin导入
1.注册激活
2.获取服务器配置信息
车辆信息收集
3.上报车辆信息
车辆信息收集
判断是否
有新版本
4.检测新版本信息
策略解析
固件下载
下载
5.下载中间结果上报
5.下载最终结果上报
车况判断
6.升级中间结果上报
汽车升级
6.升级最终结果上报
7.文件上报
8.事件上报

2.2 流程简介

车端和 OTA 平台具体交互可分为以下几个阶段：

1. OTA 平台推送升级请求/车端定期轮序/用户手动检测触发。
2. 汽车端收集 ECU 相关版本信息以及存储空间等信息上报 OTA 服务器。
3. OTA 服务器比对 ECU 版本信息，下发升级匹配策略。
4. 汽车端解析升级策略，并下载固件升级包。
5. 汽车端根据策略定义以及本地状态进行本地升级。
6. 汽车端升级完成，并上报升级结果。
7. 若升级失败，汽车端可以选择上报升级失败日志。
8. 设备端在检测版本、下载、升级过程中可以向服务器上报相关的调试日志信息。

2.3 数据类型

类型	描述
字符 String	文本信息，表明最大字符数
整型 Int	数值信息，4 字节
整型 Long	数值信息，8 字节

2.4 接口详细说明

2.4.1 注册接口

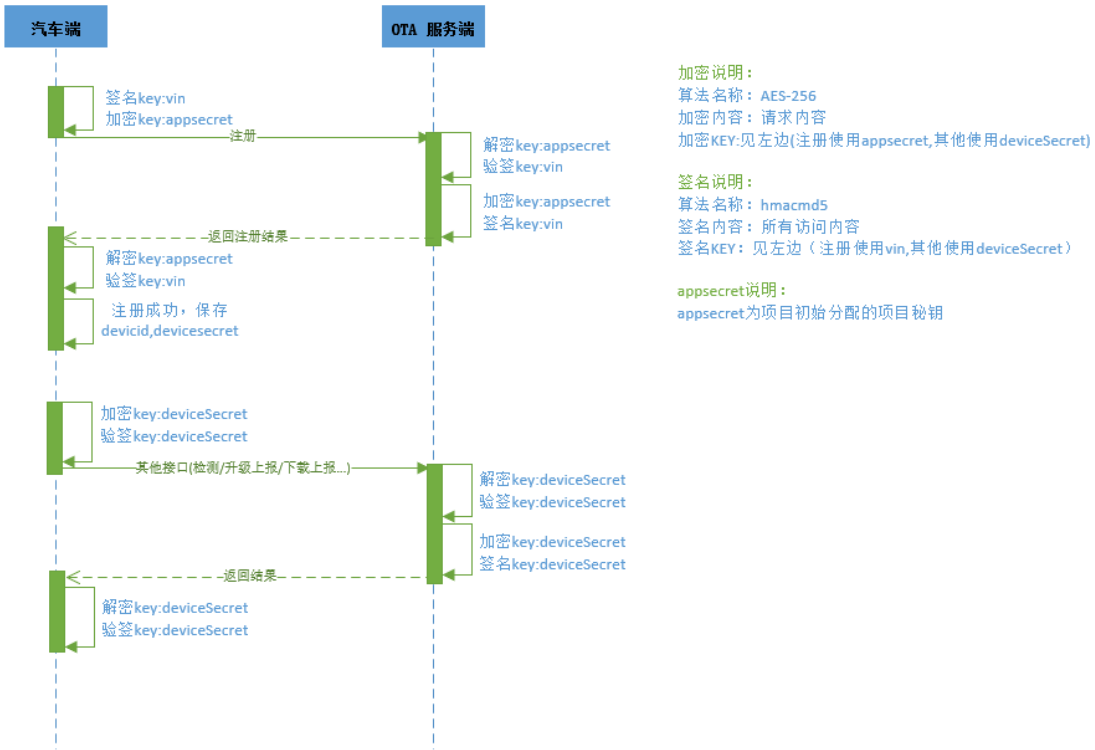
车辆注册过程由三步组成（1.注册 2.获取服务器配置信息 3.上报车辆信息）。车辆在注册时需要按照步骤依次调用这3个接口。

车辆首次激活时，需向OTA服务器注册车辆信息。车辆注册成功后，OTA服务器则分配该车辆设备ID以及设备秘钥，后续用设备ID和服务端进行交互。

接口调用时机：

1. 汽车首次注册时
2. 汽车收到2002状态码时

- **时序图**



- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/register	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- 请求参数说明

字段	类型	是否必填	备注
vin	String(17)	是	车辆唯一识别码（17 位字符串）
timestamp	Long	是	UTC 时间格式 （精确到秒，十位数）
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{vin} 算法：hmacmd5

- 请求示例(仅供参考，以请求参数说明为准)

```
{
    "vin":"interfacetest0001",
    "sdkVersion":"v1.0",
    "sign":"8ffb3feb0350ad574921c467f4922c25",
    "timestamp":"1571641944"
}
```

- 响应参数说明

字段		类型	是否必填	备注
status		Int	是	返回状态
msg		String(50)	是	状态说明
sign		String(32)	是	签名内容：参考 1.7 使用 key：{vin} 算法：hmacmd5
data	deviceId	String(32)	是	车端身份 ID（vin 的 hexMD5 值）
	secret	String(32)	是	车端消息签名 key

- 响应示例(仅供参考，以响应参数说明为准)

```
{
    "data": {
        "deviceId": "1967ab48cba77cf353d70a73aeabfdd2",
        "secret": "38da733a1dd1e81cce3ea3c18cc30409"
    },
    "msg": "成功",
    "sign": "c4c96062307ee13682ff2ded5ab683e7",
    "status": 1000
}
```

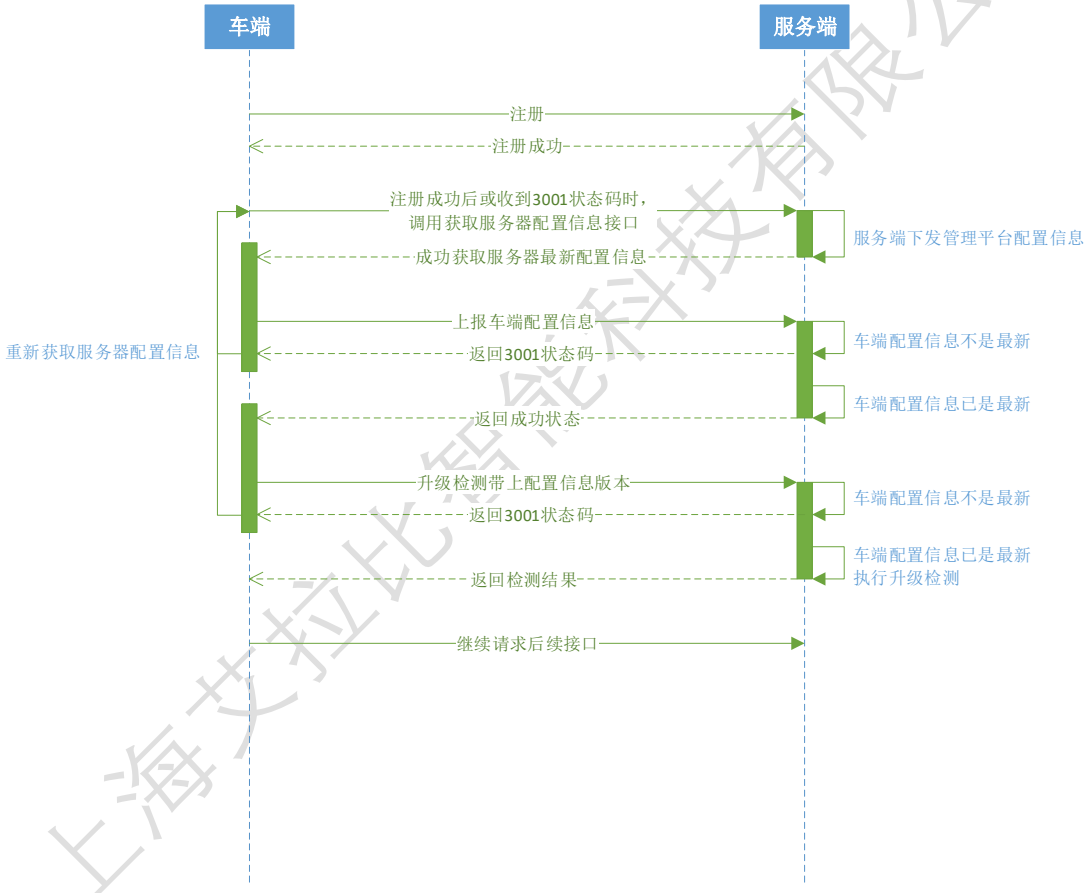
2.4.2 获取服务器配置接口

调用该接口，从服务器获取该车型对应的具有OTA能力的ECU信息。在后续ECU信息上报、检测新版本时，上报这些ECU详细信息。

接口调用时机：

1. 注册接口调用完成后
2. 汽车收到3001状态码时

- **时序图**



- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/get/server/config	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- 请求参数说明

字段	类型	是否必填	备注
timestamp	Long	是	UTC 时间格式 (精确到秒，十位数)
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5

- 请求示例(仅供参考，以请求参数说明为准)

```
{
    "sign": "c8ba4d162279ed17eb20889527ca9b5f",
    "sdkVersion": "v1.0",
    "timestamp": 1571642087
}
```

- 响应参数说明

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态说明
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
data	Object	是	返回的具体数据

data 对象内容：

confVersion	String(21)	是	服务端配置信息的版本号
ecus	List	是	服务端配置的该车型具有 OTA 能力的 ECU 信息
maxFireTrigger	Int	是	点火触发检测上限（单位：次/天）
cyclePeriod	Int	是	定期轮询周期（单位：小时）

ecus 对象内容：

ecuName	String(150)	是	零件名称
ecuDid	String(150)	是	诊断 Id
ecuSwid	String(150)	是	软件 Id
ecuResponseId	String(150)	是	响应 Id
ecuFunctionId	String(150)	是	功能 Id
installType	String(150)	是	安装类型（通过管理平台配置，下发的值为配置的**字典索引**）

- 响应示例(仅供参考，以响应参数说明为准)

```
{
    "data": {
        "confVersion": "2019-10-28 13:53:57.0",
        "cyclePeriod": 25,
        "ecus": [
            {
                "ecuDid": "D191021",
                "ecuFunctionId": "7DF",
                "ecuName": "测试零件",
                "ecuResponseId": "D191029",
                "ecuSwid": "sw191021",
                "installType": "1"
            },
            {
                "ecuDid": "USBD191021",
                "ecuFunctionId": "7DF",
                "ecuName": "测试零件",
                "ecuResponseId": "NAN",
                "ecuSwid": "usbsw191021",
                "installType": "2"
            },
            {
                "ecuDid": "RELY01",
                "ecuFunctionId": "7DF",
                "ecuName": "接口测试零件 4",
                "ecuResponseId": "NAN",
                "ecuSwid": "rely01",
                "installType": "1"
            },
            {
                "ecuDid": "15713705843",
                "ecuFunctionId": "7DF",
                "ecuName": "156947153366",
                "ecuResponseId": "1571370584B",
                "ecuSwid": "157137058450",
                "installType": "1"
            },
            {
                "ecuDid": "157137058451",
                "ecuFunctionId": "7DF",
                "ecuName": "156947153445",
                "ecuResponseId": "157137058459",
                "ecuSwid": "157137058410",
```

```
                    "installType": "1"
                }
            ],
            "maxFireTrigger": 3
        },
        "msg": "成功",
        "sign": "5e790e05b6b65ae783d301f66f570683",
        "status": 1000
}
```

2.4.3 上报汽车端信息接口

汽车调用该接口上报汽车端的相关信息

接口调用时机：

1. 车辆注册并获得服务端配置后
2. 汽车端自定义策略调用该接口
3. 消息推送

- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/report/vehicle/config	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- **请求参数说明**

字段	类型	是否必填	备注
cmdId	String(4)	是	触发方式： 1、 消息推送 2、 点火 3、 用户触发 定期轮询
timestamp	Long	是	UTC 时间格式 (精确到秒，十位数)
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
versionNumber	String(200)	否	大版本号 格式:V1.0.0(VC3SAPM5JT) **注意:该值若上报则必须和大版本序列号成对出现**
~~versionSequence~~	~~String(255)~~	~~否~~	~~大版本序列号(服务端根据该字段比较大版本大小)~~ ~~格式:X.Y.Z，其中 X/Y/Z 都是 >= 0 的整数~~ ~~**注意:该值若上报则必须和大版本号成对出现**~~
confVersion	String(21)	是	汽车端存储的服务端配置的版本号

ecus	List	是	车端具有 OTA 能力的 ECU 信息
networkInfo	Object	是	网络信息

ecus 对象内容：

ecuPartNum	String(150)	是	零件编号
ecuDid	String(150)	是	诊断 Id
ecuSwid	String(150)	是	软件 Id
ecuSVer	String(150)	是	软件版本号
ecuHsn	String(150)	是	序列号
supplierCode	String(150)	是	供应商编码
ecuHVer	String(150)	否	硬件版本
productDate	String(150)	否	生产日期（格式：yyyy-MM-dd）

networkInfo 对象内容：

netType	String(150)	是	网络接入类型，取值： 2G 3G 4G 5G WiFi Other
lac	String(150)	否	移动通信系统中的位置区码
cid	String(150)	否	客户身份
mcc	String(150)	否	移动国家号
mnc	String(150)	否	移动网络号码
rxLev	String(150)	否	接收信号电频

- **请求示例(仅供参考，以请求参数说明为准)**

```
{
    "cmdId": "3",
    "confVersion": "2019-10-28 13:53:57.0",
    "versionNumber": "V1.0.0(VC3SAPM5JT)",
    "versionSequence": "1.0.1",
    "networkInfo": {
        "mnc": "mnc1",
        "netType": "2G",
        "rxLev": "rxLev1",
        "mcc": "mcc1",
        "lac": "lac1",
        "cid": "cid1"
    },
    "ecus": [
        {
```

```
            "ecuPartNum": "pn191021",
            "ecuHVer": "v1",
            "ecuDid": "D191021",
            "ecuSwid": "sw191021",
            "ecuSVer": "v1",
            "supplierCode": "J20181221",
            "productDate": "2018-10-27",
            "ecuHsn": "sn1"
        },
        {
            "ecuPartNum": "usbpn191021",
            "ecuHVer": "v1",
            "ecuDid": "USBD191021",
            "ecuSwid": "usbsw191021",
            "ecuSVer": "v1",
            "supplierCode": "J20181221",
            "productDate": "2018-10-27",
            "ecuHsn": "sn1"
        }
    ],
    "sign": "3fd2d48c291b7bbabade73be58f45a02",
    "sdkVersion": "v1",
    "deviceId": "915c1c268d87e48f833ec1e3e3ff1730",
    "timestamp": 1561961392
}
```


- **响应参数说明**

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态说明
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5

- **响应示例(仅供参考，以响应参数说明为准)**


```
{
    "msg": "成功",
    "sign": "a55c1a01397dd6a71f1605ba866631ef",
    "status": 1000
}
```

2.4.4 检测接口

检测服务端是否有新版本。

接口调用时机：

1. 消息推送
2. 条件触发（比如说acc on）
3. 用户点击
4. 轮循

- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/check	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- **请求体参数**

同 2.4.3 上报汽车端信息接口请求体参数

- **请求示例(仅供参考，以请求参数说明为准)**

```
{
    "cmdId": "3",
    "confVersion": "2019-10-28 13:53:57.0",
    "versionNumber": "V1.0.0(VC3SAPM5JT)",
    ~~"versionSequence": "1.0.1",~~
    "networkInfo": {
        "mnc": "mnc1",
        "netType": "2G",
        "rxLev": "rxLev1",
        "mcc": "mcc1",
        "lac": "lac1",
        "cid": "cid1"
    },
    "ecus": [
        {
            "ecuPartNum": "pn191021",
            "ecuHVer": "v1",
```

```
            "ecuDid": "D191021",
            "ecuSwid": "sw191021",
            "ecuSVer": "v1",
            "supplierCode": "J20181221",
            "productDate": "2018-10-27",
            "ecuHsn": "sn1"
        },
        {
            "ecuPartNum": "usbpn191021",
            "ecuHVer": "v1",
            "ecuDid": "USBD191021",
            "ecuSwid": "usbsw191021",
            "ecuSVer": "v1",
            "supplierCode": "J20181221",
            "productDate": "2018-10-27",
            "ecuHsn": "sn1"
        }
    ],
    "sign": "3fd2d48c291b7bbabade73be58f45a02",
    "sdkVersion": "v1",
    "deviceId": "915c1c268d87e48f833ec1e3e3ff1730",
    "timestamp": 1561961392
}
```


- **响应体参数**

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态解释
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
data	Object	是	返回内容

data 对象内容：

sessionId	String(21)	是	存在新版本下发 sessionId
taskId	Long	是	任务 ID（测试车辆默认为-1）
strategyId	Long	是	策略 ID
versionNumber	String(200)	否	大版本号 当下发的任务关联的是策略时，不下发该字段；当下发的任务关联的是大版本时，下发该字段
~~versionSequence~~	~~String(255)~~	~~否~~	~~大版本序列号(客户端存储/更新该版~~

			~~本序列号，在下次检测时上报该版本序列号)~~ ~~当下发的任务关联的是策略时，不下发该字段；当下发的任务关联的是大版本时，下发该字段~~ ~~格式:X.Y.Z，其中 X/Y/Z 都是 >= 0 的整数~~
languageFileUrl	String(400)	否	多语言文件下载路径
languageFileHash	String(150)	否	多语言文件 Hash
downMaxTimes	Int	是	最大下载次数(重新下载次数)
downExpire	Int	是	下载软件包留存有效期(天数)
downNetType	Int	是	网络类型： 1：仅 wifi 2：任意网络 3：运营商网络
downNote	String(1000)	是	新版本下载提示语
downAuto	Int	是	自动下载： 1：是 2：否
upgradeMaxTimes	Int	是	最大安装次数（安装重试次数）
upgradeType	Int	是	安装类型： 1.常规安装 2 强制安装 3.静默安装
upgradeAutoStart	String(30)	否	upgradeType 为 3 时，自动安装开始时间必填
upgradeAutoEnd	String(30)	否	upgradeType 为 3 时，自动安装结束时间必填
upgradeNote	String(1000)	是	安装提示语
upgradeDisclaimer	String(4096)	否	免责申明
upgradeConditions	Object	是	升级前的车辆状态
ecusInfo	List	是	每个升级 ecu 信息
strategyTimeConsuming	Int	是	策略预计升级耗时
ecusGroupOrder	List	是	零件组、升级顺序信息

ecusInfo 对象内容：

ecuName	String(50)	是	零件名称
ecuPartNum	String(150)	是	零件编号
ecuDid	String(150)	是	诊断 id
ecuSwid	String(150)	是	软件 Id
ecuTimeConsuming	Int	否	Ecu 预计升级耗时
srcVer	String(150)	是	Ecu 当前版本
dstVer	String(150)	是	Ecu 目标版本

dependencies	List	是	Ecu 升级版本依赖
srcPkg	Object	否	原包信息
dstPkg	Object	是	目标包信息

dependencies 对象内容：

ecuPartNum	String(150)	否	依赖的 ecu
ecuDid	String(150)	是	诊断 id
ecuSwid	String(150)	是	软件 Id
ecuSVer	String(150)	是	依赖的 ecu 版本

srcPkg 和 dstPkg 对象内容：

packageType	Int	否	升级包类型： 1.整包 2.差分包
srcVer	String(150)	否	差分包的原版本/整包不需要
dstVer	String(150)	是	整包版本/差分包的目标版本
verUrl	String(355)	是	原包或加密包的下载地址；当 isEncrypt 为 0 时，表示原包下载地址；当 isEncrypt 为 1 时，表示加密包下载地址
size	Long	是	原包或加密包的大小（byte）；当 isEncrypt 为 0 时，表示原包大小；当 isEncrypt 为 1 时，表示加密包大小
hash	Object	是	包的 hash 校验码
signFileUrl	String(355)	否	签名文件下载路径
signFileSize	Long	否	签名文件大小（byte）
signFileHash	String(150)	否	签名文件 hash
description	String(255)	是	版本描述
originSize	Long	是	软件包原始大小（byte）

hash 对象内容：

isEncrypt	Int	是	是否加密；0：表示不加密；1：表示加密
oriHash	String(150)	是	原包 Hash
encHash	String(150)	否	加密包 Hash；isEncrypt 为 1 时，才会下发该字段
pkgKey	String(150)	否	加密包解密 key；isEncrypt 为 1 时，才会下发该字段

ecusGroupOrder 对象内容：

ecuPartNum	String(150)	是	零件编号
ecuDid	String(150)	是	零件诊断 id
ecuSwid	String(150)	是	零件软件 Id
upgradeOrder	String(3)	是	零件升级顺序（依据车型配置中的零

			件顺序）
groupFlag	String(5)	是	零件分组标识

upgradeConditions 对象内容（此部分内容采用平台动态配置，下表列出的前置条件仅供参考，实际请以“OTA 管理平台：配置管理->车辆条件配置”为准）：

keyInCar	Int	否	钥匙是否在车内 1:IN 2:OUT
powerMode	Int	否	电源模式 1:ON 2:OFF 3:READY 4:IGN 5:CRANK+IGN
carSpeed	Int	否	车速是否为 0 0:车速不为 0 1:车速为 0
engineSpeed	Int	否	发动机转速是否为 0 0:发动机转速不为 0 1:发动机转速为 0
motorSpeed	Int	否	电机转速是否为 0 0:电机转速不为 0 1:电机转速为 0
powerStatus	Int	否	是否为非高压、充电状态 0:高压、充电状态 1:非高压、充电状态
gearN	Int	否	档位是否为 N 档 0:非 N 档 1:N 档
gearP	Int	否	档位是否为 P 档 0:非 P 档 1:P 档
parkingStatus	Int	否.	手刹 1:UP 2:DOWN
batteryPowerMax	Int	否	蓄电池电量 最大值（%）
batteryPowerMin	Int	否	蓄电池电量 最小值（%）
batteryVoltageMax	Int	否	蓄电池电压 最大值（V）
batteryVoltageMin	Int	否	蓄电池电压 最小值（V）
batteryCurrentMax	Int	否	蓄电池电流 最大值（A）
batteryCurrentMin	Int	否	蓄电池电流 最小值（A）
motorTemperatureMin	Int	否	电机温度 最小值（°C）

motorTemperatureMax	Int	否	电机温度 最大值（°C）

- 响应示例(仅供参考，以响应参数说明为准)


```
{
    "data": {
        "downAuto": 2,
        "downExpire": 3,
        "downMaxTimes": 3,
        "downNetType": 1,
        "downNote": "0",
        "ecusGroupOrder": [
            {
                "ecuDid": "D191021",
                "ecuPartNum": "pn191021",
                "ecuSwid": "sw191021",
                "groupFlag": "65787",
                "upgradeOrder": "1"
            },
            {
                "ecuDid": "USBD191021",
                "ecuPartNum": "usbpn191021",
                "ecuSwid": "usbsw191021",
                "groupFlag": "75379",
                "upgradeOrder": "2"
            }
        ],
        "ecusInfo": [
            {
                "dependencies": [],
                "dstPkg": {
                    "description": "",
                    "dstVer": "v2",
                    "hash": {
                        "encHash":
"96220d8160f1a3bf459d963386161e78c890a672361a694a80d7bc6ec4ba4cd8",
                        "isEncrypt": 1,
                        "oriHash":
"feda25d2c28498a0322661054a36b49cc756d5c101d6117fa198ece3fd415e16",
                        "pkgKey": "86a0b84b92f84513bc8402d447e2c35a"
                    },
                    "originSize": 102400,
                    "packageType": 1,
```

```
                "signFileHash":
"af661e26e6febb8149a2e15bd2711fb2cc68e6565ade54e59fe1dcd5bc22a6d5",
                "signFileSize": 1712,
                "signFileUrl":                "https://votatest-
down.abupdate.com/download/package/654/whole/0abbfaa99d8a47e8a36a03efb78fba6c-en-sign",
                "size": 9584,
                "verUrl":                "https://votatest-
down.abupdate.com/download/package/654/whole/0abbfaa99d8a47e8a36a03efb78fba6c-en"
            },
            "dstVer": "v2",
            "ecuDid": "D191021",
            "ecuName": "测试零件",
            "ecuPartNum": "pn191021",
            "ecuSwid": "sw191021",
            "ecuTimeConsuming": 5,
            "srcVer": "v1"
        },
        {
            "dependencies": [],
            "dstPkg": {
                "description": "",
                "dstVer": "v2",
                "hash": {
                    "encHash":
"fbba762a9b22e03bfa386de2c86860af16acde3f62ed4b5221be0e660eeab080",
                    "isEncrypt": 1,
                    "oriHash":
"feda25d2c28498a0322661054a36b49cc756d5c101d6117fa198ece3fd415e16",
                    "pkgKey": "18ff2292b7ed4b42ab5e2679dd014ff9"
                },
                "originSize": 204800,
                "packageType": 1,
                "signFileHash":
"21459212cdb5ceaab0bb6edcedc86fbcc8b9602916cf093687deaca360437945",
                "signFileSize": 1712,
                "signFileUrl":                "https://votatest-
down.abupdate.com/download/package/655/whole/e218f57850b8464d9bf874fdddac6d42-en-
sign",
                "size": 9584,
                "verUrl":                "https://votatest-
down.abupdate.com/download/package/655/whole/e218f57850b8464d9bf874fdddac6d42-en"
            },
            "dstVer": "v2",
```

```
                "ecuDid": "USBD191021",
                "ecuName": "测试零件",
                "ecuPartNum": "usbpn191021",
                "ecuSwid": "usbsw191021",
                "ecuTimeConsuming": 5,
                "srcVer": "v1"
            }
        ],
        "languageFileHash":
"6438ff01d1ace921322f4bf5380bb88ee5134bc9a5ce6bb059717ca7663f4ed5",
        "languageFileUrl": "https://votatest-
down.abupdate.com/download/strategy/test/language/0d3044bdd66f4922855635d34afe3610.xml"
,
        "sessionId": "1583389054057JrLIjGOy",
        "strategyId": 308,
        "strategyTimeConsuming": 10,
        "taskId": 96,
        "upgradeAutoEnd": "",
        "upgradeAutoStart": "",
        "upgradeConditions": {
            "PowerValueMin": "1013",
            "float_testMax": "8",
            "car_1": "1",
            "car_2": "2",
            "car_3": "2",
            "PowerValue-3Min": "1",
            "speed_2": "2",
            "speed_3": "2",
            "float_testMin": "3",
            "code_testMax": "8",
            "speed_1": "1",
            "fly_1": "1",
            "fly_2": "2",
            "speed_4": "2",
            "fly_3": "2",
            "code_testMin": "3",
            "PowerValue-3Max": "20",
            "PowerValueMax": "1014"
        },
        "upgradeDisclaimer": "0",
        "upgradeMaxTimes": 3,
        "upgradeNote": "0",
        "upgradeType": 1
```

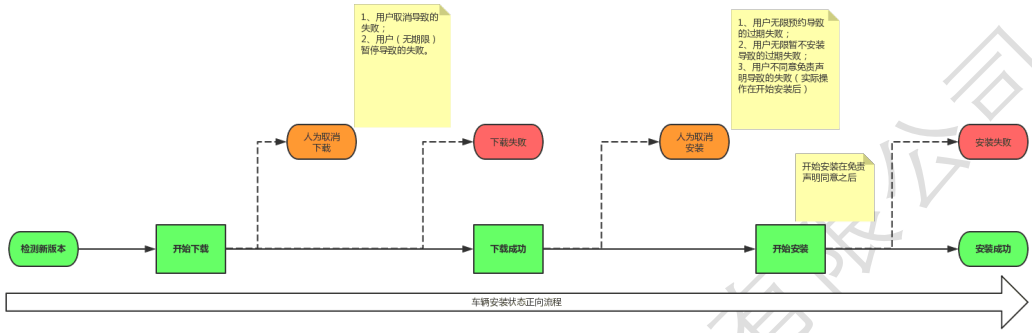
```
    },
    "msg": "成功",
    "sign": "9dce5992afceb9fe6ad37dbf753f25a9",
    "status": 1000
}
```

2.4.5 升级结果上报接口

汽车将重要的升级时间点上报给服务器。

接口调用时机：

参考该接口的resultType字段和下图



具体定义如下：

- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/vehicle/result/report	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- **请求体参数**

字段	类型	是否必填	备注
sessionId	String(21)	是	Check 接口返回的 sessionId
timestamp	Long	是	请求时间戳(精确到秒，十位数)
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
resultType	Int	是	升级阶段状态： -1.未检测（管理平台生成） 0.没有新版本（check 接口生成） 1.检测到新版本（check 接口生成） 2.下载开始 3.人为取消下载 4.下载失败

			5.下载成功 6.安装开始 7.人为取消安装 8.安装失败 9.安装成功 10.部分升级成功 11.全部升级成功 说明：-1/0/1 不需要汽车端上报；2-9 需要汽车端上报；当汽车端上报 9 时，接口会将该状态转换成 10（部分升级成功）或者 11（全部升级成功）；当汽车端上报 2/6 时，服务端会判断该任务是否已经终止，如果终止，则服务端返回状态码 6000，这是需要汽车端做相应的任务终止操作。
vehicleFailReason	String(150)	否	车辆失败原因（大状态） resultType=3/4/7/8 时，必填（参考 1.5.2）
reportEcus	Object	是	每个 ECU 的信息
versionNumber	String(200)	否	车端当前大版本号 格式:V1.0.0(VC3SAPM5JT) **注意:该值若上报则必须和大版本序列号成对出现** 如果存在大版本，只在 resultType=8/9 时，汽车端上报该值；如果不存在，则不上报
~~versionSequence~~	~~String(255)~~	~~否~~	~~大版本序列号(服务端根据该字段比较大版本大小)~~ ~~格式:X.Y.Z，其中 X/Y/Z 都是 >= 0 的整数~~ ~~**注意:该值若上报则必须和大版本号成对出现**~~ ~~如果存在大版本，只在 resultType=8/9 时，汽车端上报该值；如果不存在，则不上报~~

reportEcus 对象相关内容：

ecuPartNum	String(150)	是	Ecu 编号
ecuDid	String(150)	是	诊断 id
ecuSwid	String(150)	是	软件 Id
ecuSVer	String(150)	是	目标版本号
ecuStatus	Int	是	ecu 当前状态: 1．成功

			2．失败
ecuFailReason	String(20)	否	ECU 失败原因（小状态） ecuStatus=2 时，必填（参考 1.5.2）
beginTime	Long	否	开始升级时间(精确到秒，十位数，当 resultType 为 8 或 9 时必填)
endTime	Long	否	结束升级时间(精确到秒，十位数，当 resultType 为 8 或 9 时必填)

- **请求示例(仅供参考，以请求参数说明为准)**

```
{
    "sessionId": "1571647719440djYzSh3T",
    "timestamp": 1571644548,
    "sdkVersion": "v1.0",
    "sign": "74c84dfab3e095b1e7d7f6df65c28a62",
    "resultType": 5,
    "vehicleFailReason": "",
    "reportEcus": [
        {
            "ecuPartNum": "pn191021",
            "ecuDid": "D191021",
            "ecuSwid": "sw191021",
            "ecuSVer": "v1",
            "ecuStatus": 1,
            "ecuFailReason": ""
        },
        {
            "ecuPartNum": "usbpn191021",
            "ecuDid": "USBD191021",
            "ecuSwid": "usbsw191021",
            "ecuSVer": "v1",
            "ecuStatus": 1,
            "ecuFailReason": ""
        }
    ]
}
```

- **响应体参数**

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态
sign	String(32)	是	签名内容：参考 1.7

			使用 key：{secret} 算法：hmacmd5

- **响应示例(仅供参考，以响应参数说明为准)**

```
{
    "msg": "成功",
    "sign": "20ed0656c4fce436506bb777ce6bc5b5",
    "status": 1000
}
```

2.4.6 文件上传接口

升级日志、诊断日志等文件上传。

接口调用时机：

1. 升级文件需要上报时
2. 诊断文件需要上报时
3. 消息推送时

具体定义如下：

- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/file/upload	Post
	Header	application/x-www-form-urlencoded	
	Body	Json	
响应	Body	Json	

- **请求体参数**

字段	类型	是否必填	备注
sessionId	String(21)	是	Check 接口返回的 sessionId（如果没有，约定为字符串-1）
timestamp	Long	是	请求时间戳(精确到秒，十位数)
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
type	Int	是	文件类型 1:升级文件 2:诊断文件 3: UC 日志文件
pushId	String(100)	否	推送 id
logFile	File	是	日志文件。 文件类型：zip/log/txt/gz/可以无后缀 文件大小：不超过 128M

- **请求示例(仅供参考，以请求参数说明为准)**

```
{
    "pushId":"",
```

```
    "logFile":"/data/jenkins/workspace/autotest-public-test-4.0-cron-
api/src/test/resources/TestPackage/5881.txt",
    "sign":"9bdc95dace85df371d44dba9c5977ead",
    "sdkVersion":"v1",
    "sessionId":"1576476077271o6D2jdwc",
    "type":1,
    "deviceId":"d877f1f77ee5f13ea0f0d56e1d19e204",
    "timestamp":1576476078
}
```


- **响应体参数**

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5

- **响应示例(仅供参考，以响应参数说明为准)**


```
{
    "msg": "成功",
    "sign": "20ed0656c4fce436506bb777ce6bc5b5",
    "status": 1000
}
```

2.4.7 事件上报接口

汽车端的很多行为以事件的形式上报。

接口调用时机：

1. 事件需要上报时
2. 消息推送时

具体定义如下：

- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/event/report	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- **请求体参数**

字段	类型	是否必填	备注
sessionId	String(21)	是	Check 接口返回的 sessionId（如果没有，约定为字符串-1）
timestamp	Long	是	UTC 时间格式（精确到秒，十位数）
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
events	List	是	事件

event 对象内容：

debugLevel	Int	是	Bug 级别： 1:表示 INFO 级别 2:表示 WARN 级别 3:表示 ERROR 级别
eventId	String(150)	是	事件 ID 参考 2.8.2
subEvent	String(200)	否	子事件
eventDescription	Sring(1024)	是	事件描述（1024 个字符以内）
eventTime	Long	是	事件发生时间，UTC 时间格式（精确到秒，十位数）

- **请求示例(仅供参考，以请求参数说明为准)**

```
{
    "sessionId":"1571647719440djYzSh3T",
    "timestamp": "1571648575",
    "sdkVersion":"v1.0",
    "sign": "77028280f5e4d19cc976b272b1d98edf",
    "events":[
        {
            "debugLevel":1,
            "eventId":"1002",
            "eventTime":1571648576,
            "subEvent":"error operation",
            "eventDescription":"Desc is error. "
        }
    ]
}
```

- 响应体参数

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5

- 响应示例(仅供参考，以响应参数说明为准)

```
{
    "msg": "成功",
    "sign": "20ed0656c4fce436506bb777ce6bc5b5",
    "status": 1000
}
```

- 事件类型

Event Id	**Event Description**
1001	[phase] fsm:%d
1002	[phase] fsm:%d
1003	[phase] session:%s,taskId:%d
1004	[phase] session:%s,taskId:%d
1005	[phase] formVer:%s,newFormVer:%s
1006	[phase] ecuPartNum:%s,ecuDid:%s...

1007	[phase] result:%d
1008	[phase] result:%d
1009	[phase] freeSize:%d,totalSize:%d
1010	[phase] result:%d
1011	[phase] speed:%d
1012	[phase] ecuDid:%d,downloadStart:%d
1013	[phase] expect:%s,but:%s
1014	[phase] result:%d
1015	[phase] result:%d
1016	[phase] result:%d
1017	[phase] expect:srcVer:%s,dstVer:%s,but:srcVer:%s,dstVer:%s
1018	[phase] acc:%d,speed:%d…
1019	[phase] ecuDid:%d
1020	[phase] ecuDid:%d,result:%d

2.4.8 根证书下发接口

汽车端的根证书过期，为保证后续的安全访问，需通过该接口从服务端获取最新的根证书安装到汽车端本地。

接口调用时机：

1. 汽车端根证书即将过期前直接调用该接口
2. 汽车端根证书过期后，加上忽略证书校验调用该接口

具体定义如下：

- **接口请求和响应说明**

请求	URL	https://{ota-server}/vehicle/api/{deviceId}/ get_root_cer	Post
	Header	application/json(未加密请求报文)	text/plain(加密请求报文)
	Body	Json	
响应	Body	Json	

- **请求体参数**

字段	类型	是否必填	备注
timestamp	Long	是	UTC 时间格式（精确到秒，十位数）
sdkVersion	String(128)	是	SDK 版本号
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret}

			算法：hmacmd5
certificates	List	是	车端所有根证书列表

event 对象内容：

certificateHash	String	是	证书 Hash（不管车端根证书是否过期，都需要上传车端当前使用根证书的散列值） 算法：SHA-256
certificateFingerprint	String	是	证书指纹 算法：SHA-1

- **请求示例(仅供参考，以请求参数说明为准)**

```
{
    "certificates": [
        {
            "certificateFingerprint": "d1eb23a46d17d68fd92564c2f1f1601764d8e349",
            "certificateHash":
"d7a7a0fb5d7e2731d771e9484ebcdef71d5f0c3e0a2948782bc83ee0ea699ef4"
        }
    ],
    "sign": "4c777b3cf9f4dfc6c1db61caf6dc1428",
    "sdkVersion": "v1.0",
    "deviceId": "4f0e4062d62f2cd9e8edaa5973b4b13f",
    "timestamp": "1573178325"
}
```

- **响应体参数**

字段	类型	是否必填	备注
status	Int	是	返回状态
msg	String(50)	是	状态
sign	String(32)	是	签名内容：参考 1.7 使用 key：{secret} 算法：hmacmd5
data	Object	否	返回内容

data 对象内容：

certificates	List	是	服务端处于启用状态的证书列表

certificates 对象内容：

certificateName	String	是	证书名称
certificateContent	String	是	证书内容
certificateHash	String	是	证书 Hash 算法：SHA-256

certificateFingerprint	String	是	证书指纹 算法：SHA-1

- 响应示例(仅供参考，以响应参数说明为准)


```
{
    "data": {
        "certificates": [
            {
                "certificateContent": "-----BEGIN CERTIFICATE-----
\nMIIEsTCCA5mgAwIBAgIQCKWiRs1LXIyD1wK0u6tTSTANBgkqhkiG9w0BAQsFADBh\nM
QswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLEx
B3\nd3cuZGlnaWNlcnQuY29tMSAwHgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdC
BD\nQTAeFw0xNzExMDYxMjIzMzNaFw0yNzExMDYxMjIzMzNaMF4xCzAJBgNVBAYTAl
VT\nMRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxGTAXBgNVBAsTEHd3dy5kaWdpY2V
ydC5j\nb20xHTAbBgNVBAMTFFJhcGlkU1NMIFJTQSBDQSAyMDE4MIIBIjANBgkqhkiG9w
0B\nAQEFAAOCAQ8AMIIBCgKCAQEA5S2oihEo9nnpezoziDtx4WWLLCll/e0t1EYemE5n\n+
MgP5viaHLy+VpHP+ndX5D18INIuuAV8wFq26KF5U0WNIZiQp6mLtIWjUeWDPA28\nOeyhT
lj9TLk2beytbtFU6ypbpWUltmvY5V8ngspC7nFRNCjpfnDED2kRyJzO8yoK\nMFz4J4JE8N7NA
1uJwUEFMUvHLs0scLoPZkKcewIRm1RV2AxmFQxJkdf7YN9Pckki\nf2Xgm3b48BZn0zf0qXs
SeGu84ua9gwzjzI7tbTBjayTpT+/XpWuBVv6fvarI6bik\nKB859OSGQuw73XXgeuFwEPHTIRo
Utkzu3/EQ+LtwznkkdQIDAQABo4IBZjCCAWIw\nHQYDVR0OBBYEFPKF1n8a8ADIS8aru
SqqByCVtp1MB8GA1UdIwQYMBaAFAPeUDVW\n0Uy7ZvCj4hsbw5eyPdFVMA4GA1UdDw
EB/wQEAwIBhjAdBgNVHSUEFjAUBggrBgEF\nBQcDAQYIKwYBBQUHAwIwEgYDVR0TA
QH/BAgwBgEB/wIBADA0BggrBgEFBQcBAQQo\nMCYwJAYIKwYBBQUHMAGGGGh0dH
A6Ly9vY3NwLmRpZ2ljZXJ0LmNvbTBCBgNVHR8E\nOzA5MDegNaAzhjFodHRwOi8vY3Js
My5kaWdpY2VydC5jb20vRGlnaUNlcnRHbG9i\nYWxSb290Q0EuY3JsMGMGA1UdIARcMFo
wNwYJYIZIAYb9bAECMCowKAYIKwYBBQUH\nAgEWHGh0dHBzOi8vd3d3LmRpZ2ljZXJ
0LmNvbS9DUFMwCwYJYIZIAYb9bAEBMAgG\nBmeBDAECATAIBgZngQwBAgIwDQYJK
oZIhvcNAQELBQADggEBAH4jx/LKNW5ZklFc\nYWs8Ejbm0nyzKeZC2KOVYR7P8gevKysl
Wm4Xo4BSzKr235FsJ4aFt6yAiv1eY0tZ\n/ZN18bOGSGStoEc/JE4ocIzr8P5Mg11kRYHbmgYnr
1Rxeki5mSeb39DGxTpJD4kG\nhs5lXNoo4conUiiJwKaqH7vh2baryd8pMISag83JUqyVGc2tWP
pO0329/CWq2kry\nqv66OSMjwulUz0dXf4OHQasR7CNfIr+4KScc6ABlQ5RDF86PGeE6kdwS
QkFiB/cQ\nysNyq0jEDQTkfa2pjmuWtMCNbBnhFXBYejfubIhaUbEv2FOQB3dCav+FPg5eEve
X\nTVyMnGo=\n-----END CERTIFICATE-----",
                "certificateFingerprint": "98c6a8dc887963ba3cf9c2731cbdd3f7de05ac2d",
                "certificateHash":
"c790b47128447ec0b60f22bfcb795d71c326dd910ee12cbb4cc5a86191eb91bc",
                "certificateName": "RapidSSL.crt"
            }
        ]
    },
    "msg": "成功",
```

```
    "sign": "f7d1068950dc21ffdafbbacd8e8ebf45",
    "status": 1000
}
```