

API文档-车辆信息认证_增强版

一. 产品功能

二. 计费说明

三. 接入说明

(一) 通用接入流程

1. appcode获取

2. 主账号ID获取

(二) 接口说明

1. 输入参数

2. 输出参数

1) 成功返回示例

2) 失败返回示例

3. 字典说明

4. 接入demo

1) Java POM依赖:

2) Java代码Demo

如有问题请加产品服务钉钉群：32565068，或联系对接阿里云商务。

一. 产品功能

功能说明：通过权威实时数据源查询车辆信息。

二. 计费说明

计费逻辑：bizCode=1（一致）或者2（不一致）计费，bizCode=3（为查得）不计费

三. 接入说明

（一）通用接入流程

步骤一：打开云市场[产品链接](#)；

步骤二：点击“购买资源包”，按需购买对应规格；

步骤三：准备接口接入；

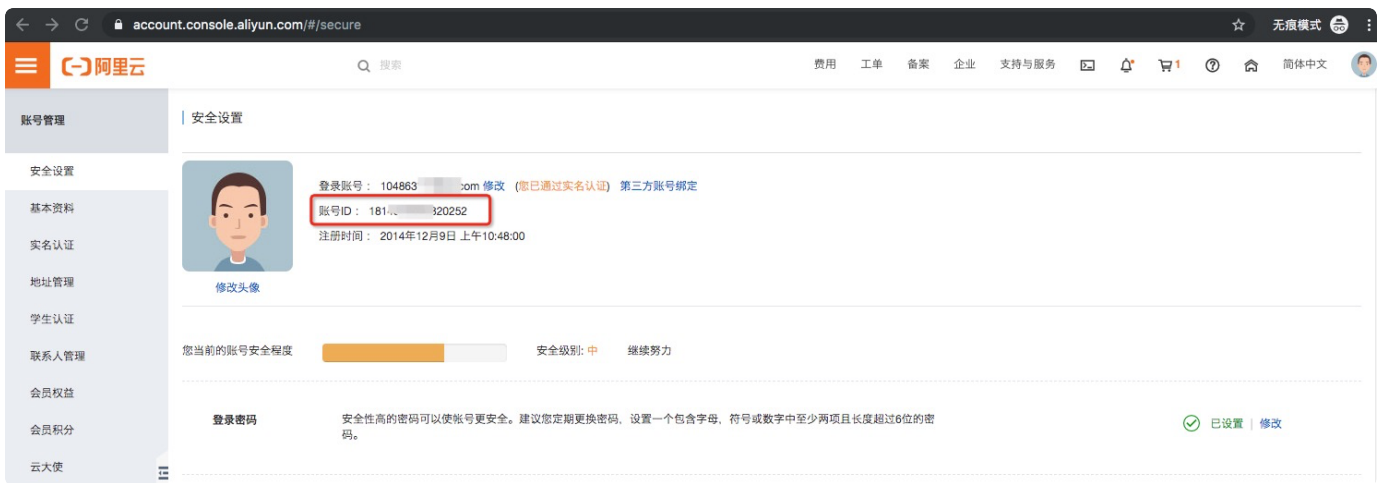
1. appcode获取

- 购买产品之后才有appcode
- <https://market.console.aliyun.com/imageconsole/index.htm?#/bizlist>



2. 主账号ID获取

- 需要主账号登录阿里云官网
- 地址：<https://account.console.aliyun.com/v2/#/basic-info/index>



(二) 接口说明

1. 输入参数

*如需加密请注意加密规则

名称	类型	是否必须	描述
productType	STRING	必选	调用产品类型，枚举值：VEHICLE_2_META为姓名+车牌号核验；VEHICLE_3_META为姓名+车牌号+身份证号
identifyNum	STRING	可选	身份证号，identifyNum和identifyNumMd5二选一
identifyNumMd5	STRING	可选	身份证号MD5，年月日加密（32位小写MD5），identifyNum和identifyNumMd5二选一
userName	STRING	可选	姓名，username和userNameMd5二选一
userNameMd5	STRING	可选	姓名MD5，第一个字加密（32位小写MD5），username和userNameMd5二选一
vehicleNum	STRING	可选	车牌号，vehicleNum和vehicleNumMd5二选一
vehicleNumMd5	STRING	可选	车牌号MD5，最后两个字母加密（32位小写MD5），vehicleNum和vehicleNumMd5二选一
vehicleType	STRING	必选	车牌类型，枚举值，02为普通小客车；52为新能源小客车

2.输出参数

参数名称	数据类型	描述
code	Integer	返回码。取值：
		200：成功
		400：参数不能为空
		401：参数非法
		402：无权限调用
		403：Api Market Subscription quota exhausted <u>购买的云市场API配额已耗尽</u> ，续费购买即可。
		403：Api Market Subscription expired API订购关系已过期
		403：异常重复调用（24小时同身份认证超过10次）
		406：需要取得用户实际授权后再调用产品
		500：系统错误
		其他状态码见API接口详情： https://help.aliyun.com/document_detail/43906.html?spm=5176.10695662.1996646101.searchclickresult.10594d961SXI3B
message	String	返回信息
requestId	String	唯一请求ID，定位问题时需提供
+data	Object	查询结果
bizCode	String	1：一致（计费） 2：不一致（计费） 3：未查得（不计费）
brand	String	品牌
vin	String	车辆识别代码
modelNum	String	车辆型号

engineNum	String	发动机号
engineType	String	发动机型号
vehicleState	String	机动车状态
approvedLoad	String	核定载客量
inspectionDate	String	检验有效期至，示例：2024-03-04 00:00:00
retirementDate	String	强制报废日期，示例：2039-03-04 00:00:00
registrationDate	String	初次登记日期，示例：2008-03-04 00:00:00
releaseDate	String	出厂日期，示例：2008-03-04 00:00:00
color	String	车身颜色
wheelBase	String	轴距
frontWheelDistance	String	前轮距
backWheelDistance	String	后轮距
displacement	String	排量
power	String	功率
axleCount	String	轴数
totalMass	String	总质量
type	String	车辆类型
unladenMass	String	整备质量
useProperty	String	使用性质
approvedCount	String	核定载客
fuelType	String	燃料种类

1) 成功返回示例

```
1 {
2   "code": "200",
3   "message": "调用成功",
4   "requestId": "06B25ED5-8D47-59C8-B9ED-1FFA53B9FB34",
5   "data": {
6     "bizCode": "1",
7     "modelNum": "SGM*****",
8     "approvedLoad": "0",
9     "color": "白",
10    "wheelBase": "2890",
11    "engineType": "LFV",
12    "totalMass": "1960",
13    "type": "小型轿车",
14    "unladenMass": "1520",
15    "useProperty": "非营运",
16    "registrationDate": "2021-**-** 00:00:00",
17    "vin": "LSGG*****",
18    "displacement": "1500",
19    "power": "120",
20    "frontWheelDistance": "1585",
21    "brand": "****",
22    "inspectionDate": "2022-**-** 00:00:00",
23    "vehicleState": "正常",
24    "backWheelDistance": "1587",
25    "retirementDate": "2022-**-** 00:00:00",
26    "releaseDate": "2021-**-** 00:00:00",
27    "axleCount": "2",
28    "engineNum": "17*****",
29    "approvedCount": "5",
30    "fuelType": "汽油"
31  }
32 }
```

备注说明：

受限于权威数据源原因，返回车辆详情字段可能为空，接口对接请做好兼容处理。

2) 失败返回示例

```
1 {  
2   "code": "400",  
3   "message": "参数不能为空(userName/userNameMd5)",  
4   "requestId": "C8C434DB-C971-4795-98F8-AFABF31FF0A2"  
5 }
```

3.字典说明

vehicleState (机动车状态)

机动车状态

正常

转出

被盗抢

停驶

注销

违法未处理

海关监管

事故未处理

嫌疑车

查封

暂扣

强制注销

事故逃逸

锁定

达到报废标准公告

牌证作废

逾期未审验

逾期三个检验周期
未检验

其他

type (车辆类型)

对应车辆类型

重型普通半挂车

重型厢式半挂车

重型罐式半挂车

重型平板半挂车

重型集装箱半挂车

重型自卸半挂车

重型特殊结构半挂车

重型仓栅式半挂车

重型旅居半挂车

重型专项作业半挂车

重型低平板半挂车

中型普通半挂车

中型厢式半挂车

中型罐式半挂车

中型平板半挂车

中型集装箱半挂车

中型自卸半挂车

中型特殊结构半挂车

中型仓栅式半挂车

中型旅居半挂车

中型专项作业半挂车
中型低平板半挂车
轻型普通半挂车
轻型厢式半挂车
轻型罐式半挂车
轻型平板半挂车
轻型自卸半挂车
轻型仓栅式半挂车
轻型旅居半挂车
轻型专项作业半挂车
轻型低平板半挂车
无轨电车
有轨电车
重型普通全挂车
重型厢式全挂车
重型罐式全挂车
重型平板全挂车
重型集装箱全挂车
重型自卸全挂车
重型仓栅式全挂车
重型旅居全挂车
重型专项作业全挂车
中型普通全挂车
中型厢式全挂车
中型罐式全挂车

中型平板全挂车
中型集装箱全挂车
中型自卸全挂车
中型仓栅式全挂车
中型旅居全挂车
中型专项作业全挂车
轻型普通全挂车
轻型厢式全挂车
轻型罐式全挂车
轻型平板全挂车
轻型自卸全挂车
轻型仓栅式全挂车
轻型旅居全挂车
轻型专项作业全挂车
重型普通货车
重型厢式货车
重型封闭货车
重型罐式货车
重型平板货车
重型集装箱车
重型自卸货车
重型特殊结构货车
重型仓栅式货车
中型普通货车
中型厢式货车

中型封闭货车
中型罐式货车
中型平板货车
中型集装箱车
中型自卸货车
中型特殊结构货车
中型仓栅式货车
轻型普通货车
轻型厢式货车
轻型封闭货车
轻型罐式货车
轻型平板货车
轻型自卸货车
轻型特殊结构货车
轻型仓栅式货车
微型普通货车
微型厢式货车
微型封闭货车
微型罐式货车
微型自卸货车
微型特殊结构货车
微型仓栅式货车
普通低速货车
厢式低速货车
罐式低速货车

自卸低速货车
仓栅式低速货车
轮式装载机械
轮式挖掘机械
轮式平地机械
大型普通客车
大型双层客车
大型卧铺客车
大型铰接客车
大型越野客车
大型轿车
大型专用客车
大型专用校车
中型普通客车
中型双层客车
中型卧铺客车
中型铰接客车
中型越野客车
中型轿车
中型专用客车
中型专用校车
小型普通客车
小型越野客车
小型轿车
小型专用客车

小型专用校车
小型面包车
微型普通客车
微型越野客车
微型轿车
普通正三轮摩托车
轻便正三轮摩托车
正三轮载客摩托车
正三轮载货摩托车
侧三轮摩托车
普通二轮摩托车
轻便二轮摩托车
三轮汽车
重型半挂牵引车
重型全挂牵引车
中型半挂牵引车
中型全挂牵引车
轻型半挂牵引车
轻型全挂牵引车
大型轮式拖拉机
小型轮式拖拉机
手扶拖拉机
手扶变形运输机
其它
大型专项作业车

中型专项作业车
小型专项作业车
微型专项作业车
重型专项作业车
轻型专项作业车
微型面包车
货运专用车
大型客车
中型客车
小型客车
微型客车

useProperty (使用性质)

使用性质
非营运
公路客运
公交客运
出租客运
旅游客运
货运
租赁
警用
消防
救护
工程救险
营转非

出租转非
教练
幼儿校车
小学生校车
初中生校车
危化品运输
中小学生校车
其他
预约出租转非
预约出租客运
营运

fuelType (燃料类型)

燃料类型
汽油
柴油
电
混合油
天然气
液化石油气
甲醇
乙醇
太阳能
混合动力
氢
生物燃料

无

其他

4.接入demo

1) Java POM依赖:

XML | 复制代码

```
1 <dependency>
2     <groupId>com.alibaba</groupId>
3     <artifactId>fastjson</artifactId>
4     <version>1.2.60</version>
5 </dependency>
6 <dependency>
7     <groupId>org.apache.httpcomponents</groupId>
8     <artifactId>httpclient</artifactId>
9     <version>4.2.1</version>
10 </dependency>
11 <dependency>
12     <groupId>org.apache.httpcomponents</groupId>
13     <artifactId>httpcore</artifactId>
14     <version>4.2.1</version>
15 </dependency>
16 <dependency>
17     <groupId>commons-lang</groupId>
18     <artifactId>commons-lang</artifactId>
19     <version>2.6</version>
20 </dependency>
```

2) Java代码Demo


```
1 import java.util.HashMap;
2 import java.util.Map;
3
4 import org.apache.http.HttpResponse;
5 import org.apache.http.util.EntityUtils;
6
7 public class Safrv2metaIdName {
8
9     public static void main(String[] args) {
10         String host = "http(s)://carownerid.market.alicloudapi.com";
11         String path = "/vehiclenmeta/";
12         String method = "GET";
13         String appcode = "你自己的AppCode";
14         Map<String, String> headers = new HashMap<String, String>();
15         //最后在header中的格式(中间是英文空格)为Authorization:APPCODE 83359****
16         *****139105
17         headers.put("Authorization", "APPCODE " + appcode);
18         Map<String, String> querys = new HashMap<String, String>();
19         //传identifyNum、userName, 或传identifyNumMd5、userNameMd5
20         querys.put("productType", "***");
21         querys.put("identifyNum", "***");
22         querys.put("****", "***");
23
24         Map<String, String> bodys = new HashMap<String, String>();
25
26     try {
27         /**
28          * 重要提示如下:
29          * HttpUtils请从
30          * https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/src/main/java/com/aliyun/api/gateway/demo/util/HttpUtils.java
31          * 下载
32          *
33          * 相应的依赖请参照
34          * https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/pom.xml
35          */
36         HttpResponse response = HttpUtils.doPost(host, path, method, headers, querys, bodys);
37         System.out.println(response.toString());
38         if(response == null || response.getStatusLine() == null){
39             // 未知异常
40             return;
41         }
42         // 获取状态码
```

```
42     int statusCode = response.getStatusLine().getStatusCode();
43     if(statusCode == 200){
44         // 成功查询, 具体信息在response的body中
45         System.out.println(EntityUtils.toString(response.getEntity
46     ()));
47     }else if(statusCode == 401 || statusCode == 400){
48         // 请求参数有误, 具体错误信息在response的body中
49         System.out.println(EntityUtils.toString(response.getEntity
50     ()));
51     }else{
52         // 其他错误类型, 请对照错误码文档查看
53     }
54 }catch (Exception e) {
55     e.printStackTrace();
56 }
}
```