

车型识别

瞿绍军

湖南师范大学信息科学与工程学院

网址

- 服务器网址
  - http://ai2.appinventor.mit.edu/
  - http://app.gzjkw.net
- 资源
  - http://appinventor.mit.edu/

车型识别

- 用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号、颜色及年份、位置信息。



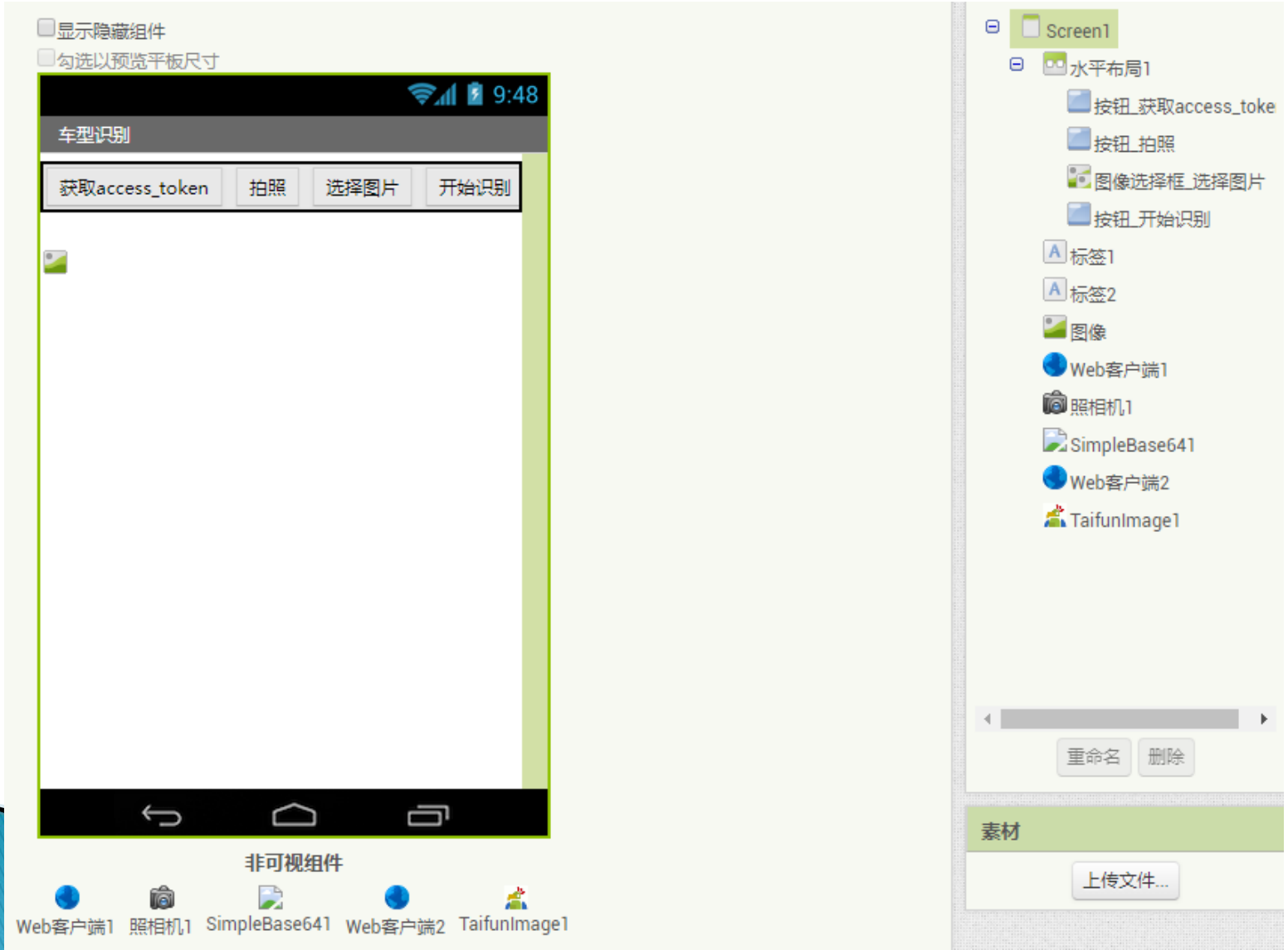


id Engineer

实现思路

- 自己训练模型，编写AIX组件
- 利用现有开放平台API（如百度、阿里云）

界面设计


显示隐藏组件
勾选以预览平板尺寸
9:48
车型识别
获取access_token
拍照
选择图片
开始识别
非可视组件
Web客户端1
照相机1
SimpleBase641
Web客户端2
TaifunImage1
Screen1
水平布局1
按钮_获取access_toke
按钮_拍照
图像选择框_选择图片
按钮_开始识别
标签1
标签2
图像
Web客户端1
照相机1
SimpleBase641
Web客户端2
TaifunImage1
重命名
删除
素材
上传文件...

界面设计

- 扩展组件
  - com.puravidaapps.TaifunImage.aix，裁剪、缩放图片
  - com.ghostfox.SimpleBase64.aix，图像数据，base64编码

利用百度AI开放平台

- https://ai.baidu.com/



注册账号


百度账号
云账号(推广帐号)
短信快捷登录
手机/邮箱/用户名
密码
登录
立即注册
忘记密码
温馨提示：
与百度搜索、百度贴吧、百度云盘、百度知道、百度文库等产品通用。
用户名
请设置用户名
设置后不可更改
中英文均可，最长14个英文或7个汉字
手机号
可用于登录和找回密码
密码
请设置登录密码
验证码
请输入验证码
获取短信验证码
阅读并接受《百度用户协议》及《百度隐私权保护声明》
注册

进入控制台


全局
图像识别
概览
应用列表
监控报表
技术文档
SDK下载
定制化图像识别
产品服务 / 图像识别 - 概览
概览
应用
已建应用: 1 个
管理应用
创建应用
用量
API
调用量
调用失败
失败率
通用物体和场景识别高级版 0 0 0%
图像主体检测 0 0 0%
logo 商标识别-入库 0 0 0%
logo 商标识别-检索 0 0 0%
logo 商标识别-删除 0 0 0%

创建应用


创建新应用
*应用名称：
*应用类型：
游戏娱乐
*接口选择：
您可以为应用勾选多个接口权限，使此应用可以请求已勾选的接口服务
图像识别
通用物体和场景识别高级版
图像主体检测
logo 商标识别
菜品识别
车型识别
动物识别
植物识别
百度语音
文字识别
人脸识别
自然语言处理
内容审核
UNIT
知识图谱
智能呼叫中心
图像搜索
人体分析
*应用描述：
简单描述一下您使用人工智能服务的应用场景，如开发一款美颜相机，需要检测人脸关键点，请控制在500字以内

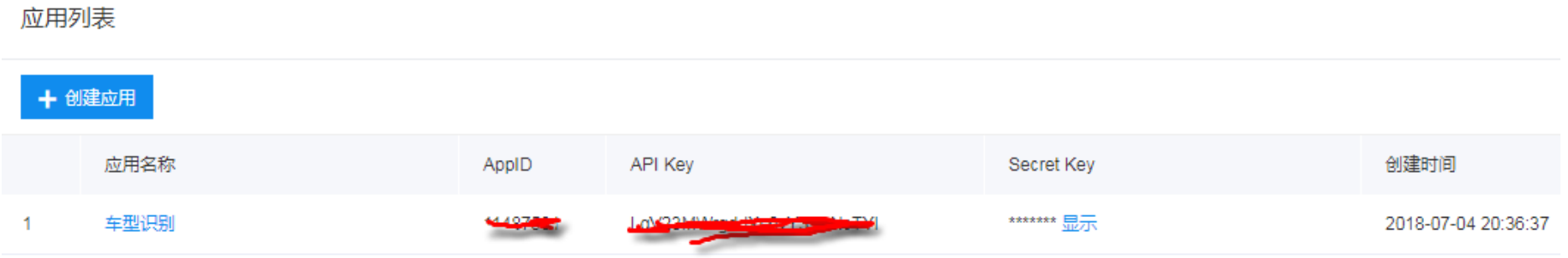
创建应用


* 应用名称：
车型识别
* 应用类型：
游戏娱乐
* 接口选择：
您可以为应用勾选多个接口权限，使此应用可以请求已勾选的接口服务
图像识别
通用物体和场景识别高级版
图像主体检测
logo 商标识别
菜品识别
车型识别
动物识别
植物识别
百度语音
文字识别
人脸识别
h5语音验证码
h5活体视频分析
人脸检测
人脸对比
人脸搜索
人脸库管理
在线活体检测
自然语言处理
内容审核
UNIT
知识图谱
智能呼叫中心
图像搜索
人体分析
* 文字识别包名：
需要
不需要
* 应用描述：
开发一款车型识别软件

创建应用

- 创建完毕后返回到应用列表

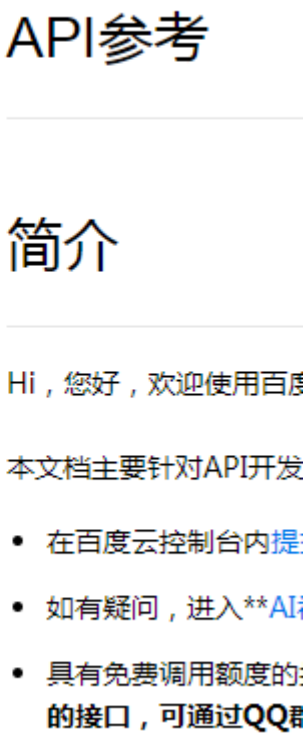




查看调用文档


全局
图像识别
概览
应用列表
监控报表
技术文档
SDK下载
定制化图像识别
产品服务 / 图像识别 - 应用列表
应用列表
创建应用
应用名称
1
车型识别
2
test


新手指南
百度语音
视觉技术
文字识别
人脸识别
图像审核
图像识别
API文档
SDK文档
产品价格


API参考
简介
Hi，您好，欢迎使用百
本文档主要针对API开发
在百度云控制台内提
如有疑问，进入**AI
具有免费调用额度的
的接口，可通过QQ

查看API文档



细粒度图像识别—车型识别

接口描述

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号、颜色及年份、位置信息。

请求说明

请求示例

HTTP 方法： `POST`

请求URL： `https://aip.baidubce.com/rest/2.0/image-classify/v1/car`

通过API Key和Secret Key获取的access_token

- 针对HTTP API调用者，百度AIP开放平台使用OAuth2.0授权调用开放API，调用API时必须在URL中带上accesss_token参数，获取Access Token的流程如下：
- http://ai.baidu.com/docs#/Auth/top

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

获取Access Token

- 请求URL数据格式
- 向授权服务地址https://aip.baidubce.com/oauth/2.0/token发送请求（推荐使用POST），并在URL中带上以下参数：
  - grant_type： 必须参数，固定为client_credentials；
  - client_id： 必须参数，应用的API Key；
  - client_secret： 必须参数，应用的Secret Key；

获取Access Token

- https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=Va5yQRHlA4Fq5eR3LT0vuXV4&client_secret=0rDSjzQ20XUj5itV6WRtznPQSzr5pVw2&

获取Access Token

- 代码



获取Access Token

- 服务器返回的JSON文本参数如下：
  - access_token： 要获取的Access Token；
  - expires_in： Access Token的有效期(秒为单位，一般为1个月)；
  - 其他参数忽略，暂时不用;

例如：


```
{
  "refresh_token": "25.b55fe1d287227ca97aab219bb249b8ab.315360000.1798284651.282335-8574074",
  "expires_in": 2592000,
  "scope": "public wise_adapt",
  "session_key": "9mzdDZXu3dENdFZQurfg0Vz8slgSgvvOAUebNFzyzcpQ5EnbxbF+hfG9DQkpUVQdh4p6HbQcAiz5RmuBAja1JJGgIdJI",
  "access_token": "24.6c5e1ff107f0e8bcef8c46d3424a0e78.2592000.1485516651.282335-8574074",
  "session_secret": "dfac94a3489fe9fca7c3221cbf7525ff"
}
```

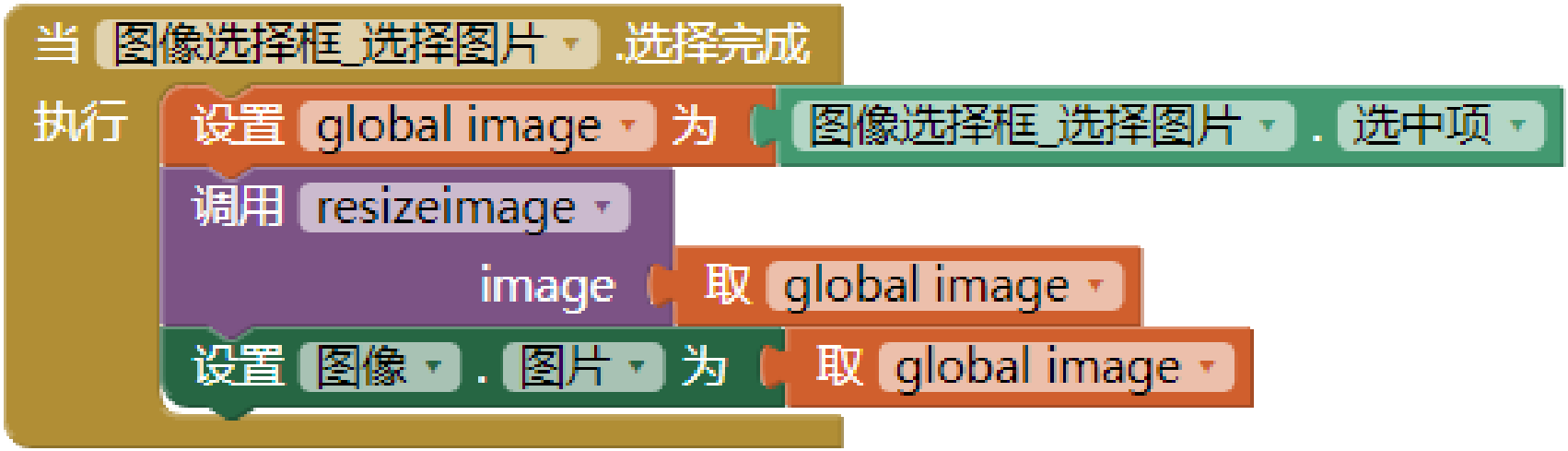
解码JSON数据


当 Web客户端1 .获得文本
URL网址 响应代码 响应类型 响应内容
执行 如果 取 响应代码 = 200
则 设置 global access_token_list 为 调用 Web客户端1 .解码JSON文本
JSON文本 取 响应内容
设置 global access_token 为 在键值对 取 global access_token_list
中查找关键字 " access_token "
, 如未找到则返回 " not found "
设置 标签1 . 文本 为 取 global access_token

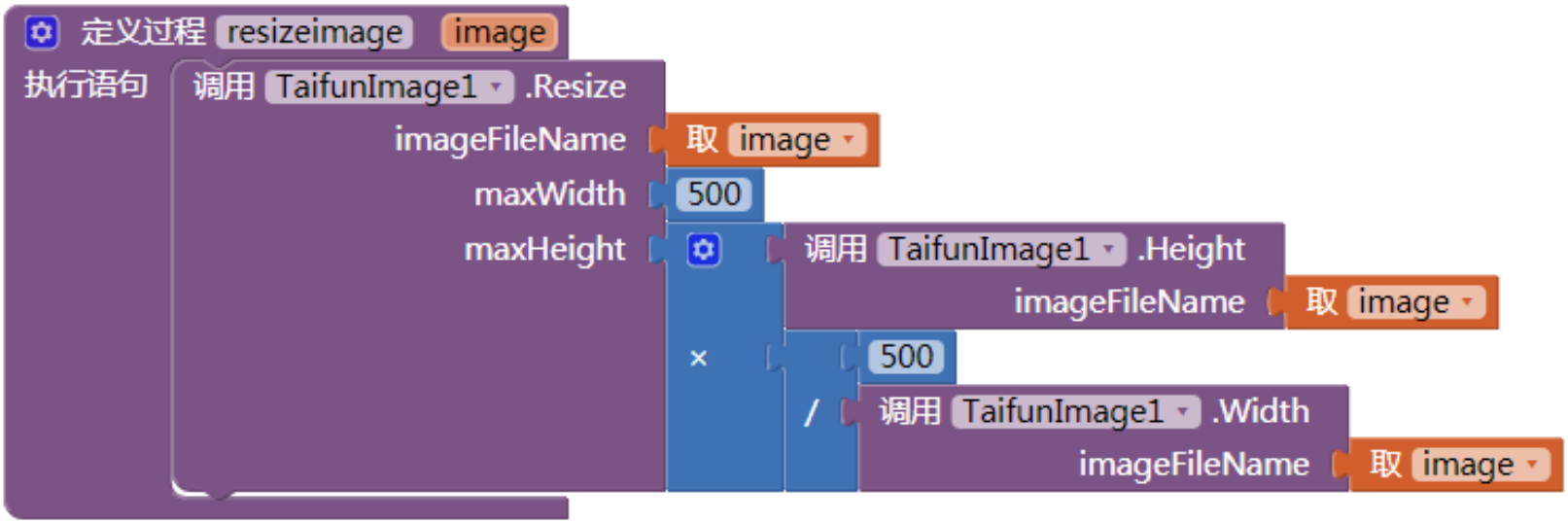
拍照


当 按钮_拍照 .被点击
执行 调用 照相机1 .拍照
当 照相机1 .拍摄完成
图像位址
执行 设置 global image 为 取 图像位址
调用 resizeimage
image 取 global image
设置 图像 . 图片 为 取 global image

选择图片


当 图像选择框_选择图片 .选择完成
执行 设置 global image 为 图像选择框_选择图片 . 选中项
调用 resizeimage
image 取 global image
设置 图像 . 图片 为 取 global image

缩放图片


定义过程 resizeimage image
执行语句
调用 TaifunImage1 .Resize
imageFileName 取 image
maxWidth 500
maxHeight
调用 TaifunImage1 .Height
imageFileName 取 image
×
500
/
调用 TaifunImage1 .Width
imageFileName 取 image

调用API接口

请求说明

- HTTP 方法：POST
- 请求URL：
  https://aip.baidubce.com/rest/2.0/image-classify/v1/car
- URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

调用API接口

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

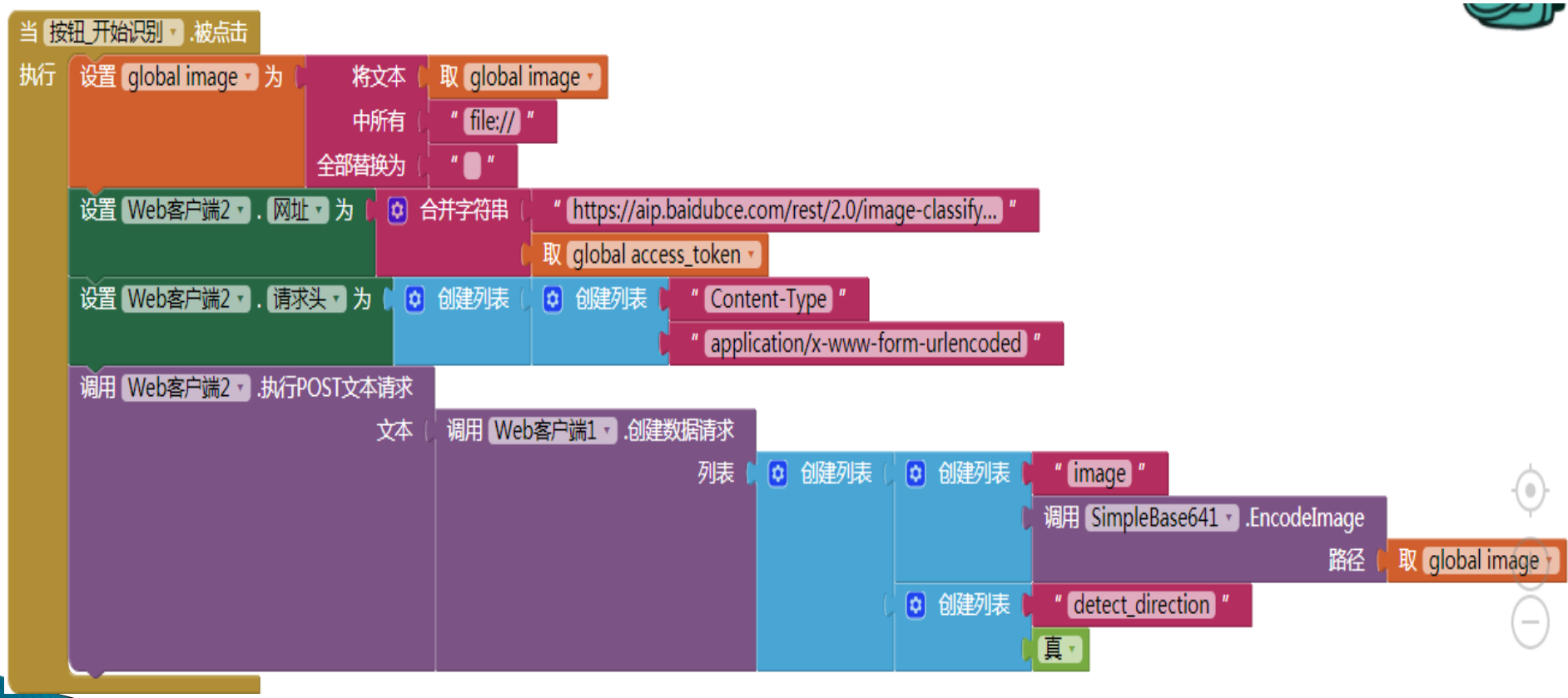
参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式
top_num	false	uint32	-	返回结果top n，默认5。

调用API接口

请求代码示例

- 车型识别 curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/car?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需UrlEncode】&top_num=5' -H 'Content-Type:application/x-www-form-urlencoded'

调用API接口


当 按钮_开始识别 .被点击
执行
设置 global image 为
将文本 取 global image
中所有 " file:// "
全部替换为 " "
设置 Web客户端2 . 网址 为
合并字符串 " https://aip.baidubce.com/rest/2.0/image-classify... "
取 global access_token
设置 Web客户端2 . 请求头 为
创建列表
创建列表
" Content-Type "
" application/x-www-form-urlencoded "
调用 Web客户端2 .执行POST文本请求
文本
调用 Web客户端1 .创建数据请求
列表
创建列表
创建列表
" image "
调用 SimpleBase641 .EncodeImage
路径
取 global image
创建列表
" detect_direction "
真

返回数据

- **返回参数**

字段	是否必选	类型	说明
log_id	否	uint64	唯一的log id，用于问题定位
color_result	是	string	颜色
result	否	car-result()	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	double	置信度，示例：0.5321
+year	否	string	年份
location_result	否	string	车在图片中的位置信息

返回数据

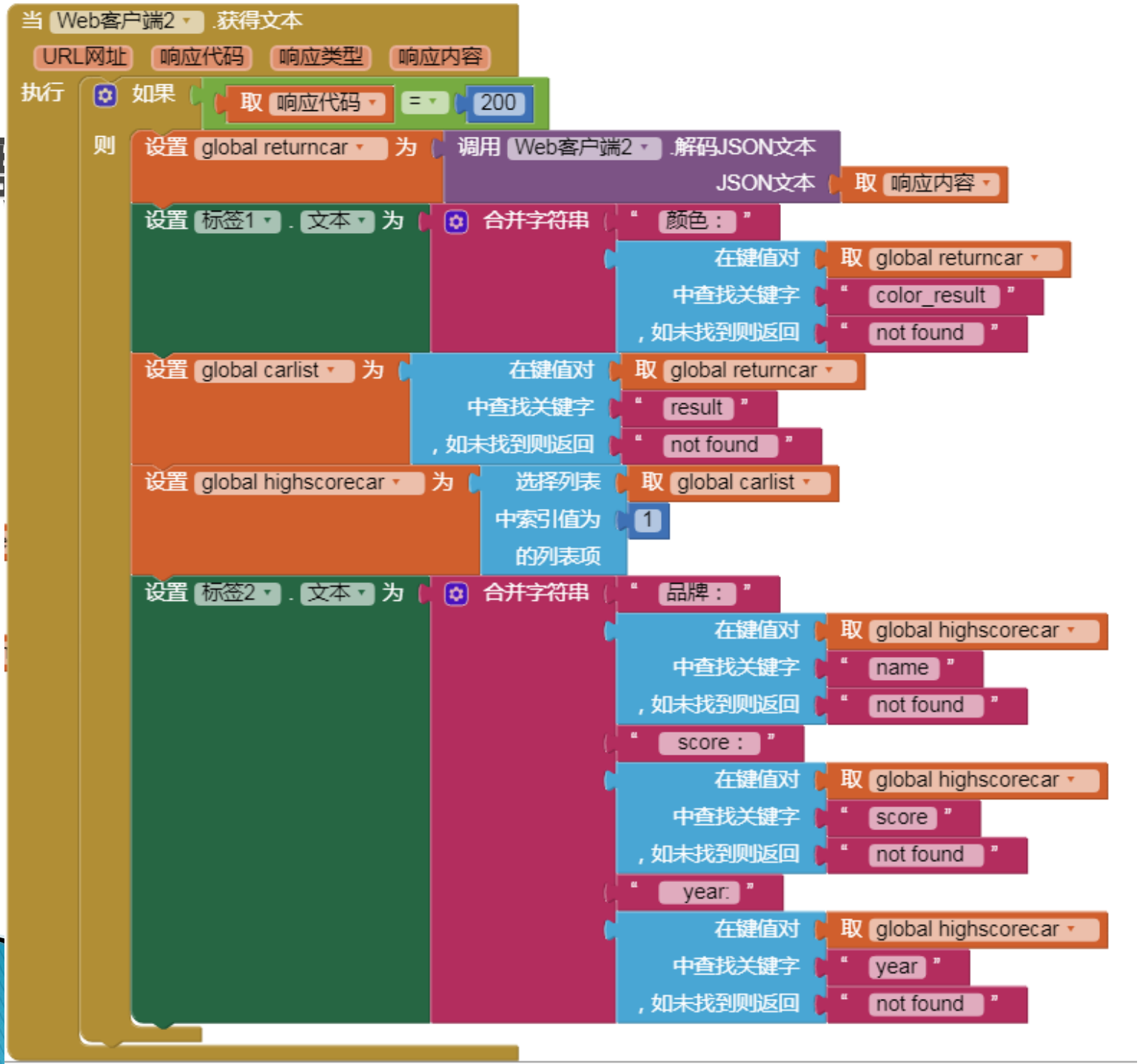
```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
        "log_id": 3490021330,
        "result": [{
                "name": "别克昂科威",
                "score": 0.83903276920319,
                "year": "2014-2017"
        },
        {
                "name": "通用(别克昂科拉)Mokka",
                "score": 0.010498280636966,
                "year": "2017"
        },
        {
                "name": "奔驰gla级",
                "score": 0.0083939796313643,
        {
                "name": "MG锐腾",
                "score": 0.0065533421002328,
                "year": "2016-2017"
        }],
        "color_result": "棕色",
        "location_result":
        {
                "left": 257,
                "top": 663,
                "width": 1229,
                "height": 795
        }
}
```

处理返回数据

- returncar：保存解码JSON后的数据
- carlist：保存返回车型结果
- highscorecar：保存可能性最高的结果



调


当 Web客户端2 .获得文本
URL网址 响应代码 响应类型 响应内容
执行 如果 取 响应代码 = 200
则 设置 global returncar 为 调用 Web客户端2 .解码JSON文本
JSON文本 取 响应内容
设置 标签1 . 文本 为 合并字符串 " 颜色： "
在键值对 取 global returncar
中查找关键字 " color_result "
，如未找到则返回 " not found "
设置 global carlist 为 在键值对 取 global returncar
中查找关键字 " result "
，如未找到则返回 " not found "
设置 global highscorecar 为 选择列表 取 global carlist
中索引值为 1
的列表项
设置 标签2 . 文本 为 合并字符串 " 品牌： "
在键值对 取 global highscorecar
中查找关键字 " name "
，如未找到则返回 " not found "
" score： "
在键值对 取 global highscorecar
中查找关键字 " score "
，如未找到则返回 " not found "
" year: "
在键值对 取 global highscorecar
中查找关键字 " year "
，如未找到则返回 " not found "

参考教材

- App Inventor移动应用开发标准教程，
- 瞿绍军
- 978-7-115-42681-9
- 人民邮电出版社

