

阿里绿网

# 内容检测API文档



# 内容检测API文档

## 简介

内容检测API基于阿里巴巴多年的技术沉淀和海量的数据支撑，提供文本、图片、视频等多媒体内容安全检测的接口服务。

## 智能鉴黄服务

通过人工智能技术和实时更新的亿级图像样本库,智能鉴黄可对用户上传的图片进行识别以及色情程度的打分,方便用户对色情图片作快速处理,帮助用户减少审核人力,有效降低涉黄风险

## 图文识别服务

拥有国内顶尖OCR算法团队,上亿字符样本积累,可精准定位图片中文字位置,准确识别文字内容

## 暴恐敏感图像识别（可申请免费试用）

高速图搜技术已经服务于阿里巴巴集团内百余条业务线.结合独有的情报、舆情、预警和分析体系及实时更新的样本图库,能够快速定位目标图像

## 文本过滤服务（可申请免费试用）

背靠阿里巴巴十余年安全领域词库沉淀,可对目标文本进行精确过滤,并快速识别风险类型

## API使用须知

欢迎使用阿里云内容检测API。用户可以使用本文档介绍的 API对内容检测API服务进行相关操作。

### 1. 阅读使用协议和收费方式

请确保在使用这些接口前,已充分了解了内容检测API的使用协议和收费方式。

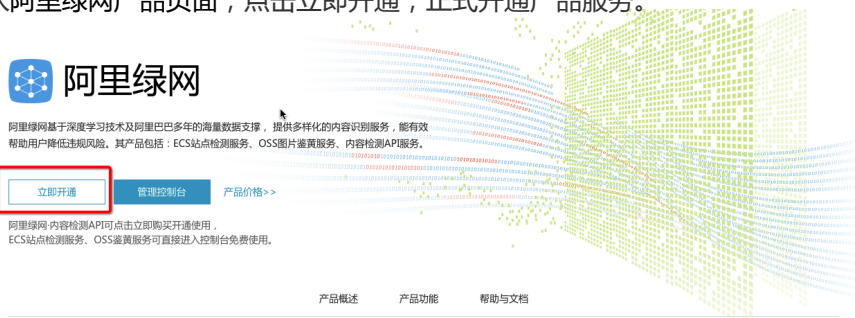
## 2. 内容检测API开发准备

开发者使用API之前，需要先注册阿里云账号，添加Access Key并签约绿网，具体操作如下：

- 前往阿里云官网注册账号，如果已有注册账号，请跳过此步骤；



- 请进入阿里绿网产品页面，点击立即开通，正式开通产品服务。



- 在Access Key管理页面中创建并管理您的Access Key，Access Key相当于您访问阿里云产品的口令，拥有您完整的权限，请您妥善保管、避免泄露，并定期更换您的Access Key，同时绿网API也支持RAM子账号的Access Key，RAM子账号请授权给系统策略:AliyunYundunGreenWebFullAccess。如果已经添加过密钥，请跳过此步骤。



注：

1. 每个用户每分钟默认最高可调用100次，如需调整，请通过工单联系我们；
2. 如果您的海量的图片需要扫描，可以通过工单联系我们，获取更加优惠的价格；

3. 如果您需要更多的内容检测服务，也可以通过工单联系我们。
4. 用户可通过智能鉴黄测试页面接入业务图片进行测试。

## 访问控制

### 用户签名验证 ( Authentication )

内容检测API通过使用AccessKeyId/ AccessKeySecret对称加密的方法来验证某个请求的发送者身份。AccessKeyId用于标示用户，AccessKeySecret是用户用于加密签名字符串和OSS用来验证签名字符串的密钥，其中AccessKeySecret必须保密，只有用户和内容检测API知道。

- 阿里云账户AccessKey：每个阿里云账号提供的AccessKey对拥有的资源有完全的权限

绿网开放服务接口使用http协议暴露。用户在访问时需要提供其在阿里云网站获得的Access Key ID，以及对应的Access Key Secret。具体的请求参数组装与加密，请参见<http://develop.aliyun.com/api/>

## 调用方式

内容检测API提供JAVA、PHP、python 3种接入方式。

### 请先阅读阿里云SDK 各环境准备指南

<https://develop.aliyun.com/sdk/java?spm=5176.1971658.101.2.MGWDe3>

#### JAVA接入

1. 建立maven工程；
2. 导入绿网开放接口依赖(aliyun-java-sdk-green)

```
<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-green</artifactId>
<version>1.4.0</version>
</dependency>

<dependency>
<groupId>com.aliyun</groupId>
<artifactId>aliyun-java-sdk-core</artifactId>
<version>2.4.2</version>
```

```
</dependency>
```

1. 参考java sdk使用介绍及示例

[https://help.aliyun.com/document\\_detail/lvwang/APIguide/SDKguide/javasdk.html](https://help.aliyun.com/document_detail/lvwang/APIguide/SDKguide/javasdk.html)

## PHP接入

1. 浏览阿里云POP在github上的PHP文档；
2. 或者直接下载依赖的sdk包:

- aliyun-php-sdk-core
- aliyun-php-sdk-green

1. 建立PHP工程；

引入aliyun-php-sdk-core与aliyun-php-sdk-green

参考php sdk使用介绍及示例

[https://help.aliyun.com/document\\_detail/lvwang/APIguide/SDKguide/phpsdk.html](https://help.aliyun.com/document_detail/lvwang/APIguide/SDKguide/phpsdk.html)

## python接入

- 1.使用pip安装sdk:

```
sudo pip install aliyun-python-sdk-core
sudo pip install aliyun-python-sdk-green
```

1. 参考python sdk使用介绍及示例

[https://help.aliyun.com/document\\_detail/lvwang/APIguide/SDKguide/pythonsdk.html](https://help.aliyun.com/document_detail/lvwang/APIguide/SDKguide/pythonsdk.html)

## 更新历史

**API版本：aliyun-java-sdk-green 1.1.0**

**最近更新时间：2016-03-08**

发布时间	更新	说明
2016-06-21	aliyun-java-sdk-green	版本更新至1.4.0

2016-03-08	aliyun-java-sdk-green	版本更新至1.1.0
2016-02-24	aliyun-java-sdk-green	版本更新至1.0.3
2016-01-18	aliyun-java-sdk-green	版本更新至1.0.2
2015-12-22	aliyun-java-sdk-green	版本更新至1.0.1
2015-12-16	aliyun-java-sdk-green	版本1.0.0正式上线

## 智能图片鉴黄API

### 图片鉴黄同步调用接口

#### 描述

- 接收用户图片并检测，并同步返回检测结果。
- 为保障同步调用的时效性，同步调用接口支持单张调用，适用于对时效性要求较高的使用场景，如：社交等。

#### 接口说明

协议	url	方法	参数
http	green.aliyuncs.com/rpc/image/detection	POST	JSON字符串

#### 请求参数

属性	值类型	是否必须	说明
image_url	List<String>	否	需要检测的图像URL，目前只支持http协议的图像URL。图片只支持静态图片格式如：PNG, JPG, JPEG, BMP，不支持GIF图片
async	boolean	否	默认同步。同步接口一次只能上传单张图像url

scene	List<String>	是	场景ID，不同场景代表使用不同算法。目前支持porn（色情图像识别）
-------	--------------	---	------------------------------------

## 返回参数

- 请求参数：JSON字符串

属性	值类型	值示例	描述
code	String	Success	返回状态码，Success表示调用成功。如果不为Success则表示调用出错，错误消息可以参考msg
msg	String	Request Success	返回消息。如果code不为Success的时候，描述详细错误信息
images	List<Object>		图片检测结果列表，每个元素是一个图片的检测结果,详见下面图片Object的描述

- 图片Object描述：

属性	值类型	说明
task_id	String	图像检测的任务ID
image_name	String	图像的URL
porn	Object	色情图像识别结果，json对象。包含两个key-value对：{"label":1, "rate":100}。rate是0~100范围的一个浮点值，越接近100，表示色情图像的概率越高（精度到小数点后2位）。label是建议值，0表示正常，1表示色情，2表示未确认。用户可以根据自己的场景采信这个值（注：绿网会根据图片样本的不断积累，动态调整建议值的阈值设定）

## 图片鉴黄异步调用接口

## 描述

- 接收用户图片并检测，返回是否接受请求成功，并返回任务ID，供用户使用异步图像检测结果查询接口来查询结果。
- 异步调用接口支持多张图像同时调用，用户可隔相对较长的时间使用图片鉴黄异步检测结果查询接口来取结果。

## 接口说明

协议	url	方法	参数
http	green.aliyuncs.com/rpc/image/detection	POST	JSON字符串

## 请求参数

属性	值类型	是否必须	说明
image_url	List<String>	否	需要检测的图像URL，目前只支持http协议的图像URL。支持多张图像上传，最大支持50张图像。图片只支持静态图片格式如：PNG,JPG,JPEG,BMP，不支持GIF图片
async	boolean	否	异步调用，请将参数置为true。
scene	List<String>	是	场景ID，不同场景代表使用不同算法。目前支持porn（色情图像识别）

## 返回参数

- 请求参数：JSON字符串

属性	值类型	值示例	说明
code	String	Success	返回状态码，Success表示调用成功。如果不为Success则表示调用出错，错误消息可以参考msg
msg	String	Request Success	返回消息。如果



			code不为Success的时候，描述详细错误信息
images	List<Map<String, String>>		图片异步检测接收列表。每个元素是一个图片的接收结果(map对象)，包含两个key-value对 : {"image_name": "imgName1", "task_id": "babac802-195d-4406-9cf7-38d71e516ed7-1450776115827"}。 image_name是图片名字，task_id是图片检测任务id(根据该id从读取图片检测结果接口获取检测结果)

注: 异步图像检测接口将会正对每一张检测的图片返回一个task\_id, 需要根据这个task\_id调用获取异步图像检测结果接口来获取图片检测的最终结果, 请参考: [图片鉴黄异步检测结果获取接口](#) 的说明

## 图片鉴黄异步检测结果查询接口

### 描述

使用异步调用接口的用户，可使用本接口来查询检测结果。

### 接口说明

协议	url	方法	参数
http	green.aliyuncs.com/rpc/image/result	GET	字符串

### 请求参数

属性	值类型	是否必须	说明
taskid	String	是	值为图片检测接口异步调用时返回的图片task_id

## 返回参数

- 请求参数：JSON字符串

属性	值类型	值示例	说明
code	String	Success	返回状态码 ，Success表示调用成功。如果不为Success则表示调用出错，错误消息可以参考msg
msg	String	Request Success	返回消息。如果code不为Success的时候，描述详细错误信息
status	String	TaskProcessSuccess	返回状态值 ，TaskProcessSuccess表示图片异步api请求task状态, 处理成功 ；TaskProcessing表示图片异步api请求task状态, 正在处理中 ；TaskInvalid表示图片异步api请求task状态, 无效task或者task不存在 ；TaskProcessFailed表示图片异步api请求task状态, 处理失败
image	object		参见image object描述

- 图片Object描述：

属性	值类型	说明
task_id	String	图像检测的任务ID
image_name	String	图像的URL
porn	Object	色情图像识别结果，json对象。包含两个key-value对： {"label":1, "rate":100}。 rate是0~100范围的一个浮点值，越接近100，表示色情图像的概率越高（精度到小数点后2位）。label是建议值，0表示正常，1表示色情，2表示未确认。用户可以根据自己的场景采信这个值（注：绿网会根据图片样本的不断积累，动态调整建议值的阈值设定）

# OCR图文识别API

## 图文识别同步调用接口

### 描述

- 接收用户图片并检测，并同步返回检测结果。
- 为保障同步调用的时效性，同步调用接口支持单张调用，适用于对时效性要求较高的使用场景，如：社交等。

### 接口说明

协议	url	方法	参数
http	green.aliyuncs.com/rpc/image/detection	POST	JSON字符串

### 请求参数

属性	值类型	是否必须	说明
image_url	List<String>	否	需要检测的图像URL，目前只支持http协议的图像URL。图片只支持静态图片格式如：PNG,JPG,JPEG,BMP，不支持GIF图片
async	boolean	否	默认同步。同步接口一次只能上传单张图像url
scene	List<String>	是	场景ID，不同场景代表使用不同算法。目前支持ocr（图文识别）

### 返回参数

- 请求参数：JSON字符串

属性	值类型	值示例	描述
code	String	Success	返回状态码，Success表示调用成功。如果不为Success则表示调用出错，错误消息可以参考msg
msg	String	Request Success	返回消息。如果code不为Success的时候，描述详细错误信息
images	List<Object>		图片检测结果列表，每个元素是一个图片的检测结果,详见下面图片Object的描述

- 图片Object描述：

属性	值类型	说明
task_id	String	图像检测的任务ID
image_name	String	图像的URL
ocr	Object	图像字符识别结果，json对象。包含一个key-value对： {"text":["图片中的文字"]} text是字符串列表，内容是图像中文字的主要段落内容（按照概率输出最多5组概率最大的组合）

## 图文识别异步调用接口

### 描述

- 接收用户图片并检测，返回是否接受请求成功，并返回任务ID，供用户使用异步图像检测结果查询接口来查询结果。
- 异步调用接口支持多张图像同时调用，用户可隔相对较长的时间使用图文识别异步检测结果查询接口来取结果。

### 接口说明

协议	url	方法	参数
----	-----	----	----

http	green.aliyuncs.com/rpc/image/detection	POST	JSON字符串
------	--	------	---------

## 请求参数

属性	值类型	是否必须	说明
image_url	List<String>	否	需要检测的图像URL，目前只支持http协议的图像URL。支持多张图像上传，最大支持50张图像。图片只支持静态图片格式如： ：PNG,JPG,JPEG,BMP，不支持GIF图片
async	boolean	否	异步调用，请将参数置为true。
scene	List<String>	是	场景ID，不同场景代表使用不同算法。目前支持ocr（图文识别）

## 返回参数

- 请求参数：JSON字符串

属性	值类型	值示例	说明
code	String	Success	返回状态码，Success表示调用成功。如果不为Success则表示调用出错，错误消息可以参考msg
msg	String	Request Success	返回消息。如果code不为Success的时候，描述详细错误信息
images	List<Map<String, String>>		图片异步检测接收列表。每个元素是一个图片的接收结果(map对象)，包含两个key-value对： ：{"image_name":"imgName1", "task_id":"babac802-195d-4406-9cf7-38d71e516ed7-1450776115827"}。image_name是图片

			名字, task_id是图片检测任务id(根据该id从读取图片检测结果接口获取检测结果)
--	--	--	--

注: 异步图像检测接口将会正对每一张检测的图片返回一个task\_id, 需要根据这个task\_id调用获取异步图像检测结果接口来获取图片检测的最终结果, 请参考: [图文识别异步检测结果获取接口](#) 的说明

## 图文识别异步检测结果查询接口

### 描述

使用异步调用接口的用户, 可使用本接口来查询检测结果。

### 接口说明

协议	url	方法	参数
http	green.aliyuncs.com/rpc/image/result	GET	字符串

### 请求参数

属性	值类型	是否必须	说明
taskid	String	是	值为图片检测接口异步调用时返回的图片task_id

### 返回参数

- 请求参数: JSON字符串

属性	值类型	值示例	说明
code	String	Success	返回状态码, Success表示调用成功。如果不为Success则表示调用出错, 错误消息可以参考msg
msg	String	Request Success	返回消息。如果code不为Success的

			时候, 描述详细错误信息
status	String	TaskProcessSuccess	返回状态值 ; TaskProcessSuccess表示图片异步api请求task状态, 处理成功 ; TaskProcessing表示图片异步api请求task状态, 正在处理中 ; TaskInvalid表示图片异步api请求task状态, 无效task或者task不存在 ; TaskProcessFailed表示图片异步api请求task状态, 处理失败
image	object		参见image object描述

- 图片Object描述 :

属性	值类型	说明
task_id	String	图像检测的任务ID
image_name	String	图像的URL
ocr	Object	图像字符识别结果, json对象。包含一个key-value对: {"text":["图片中的文字"]}。text是字符串列表, 内容是图像中文字的主要段落内容(按照概率输出最多5组概率最大的组合)

## SDK使用手册

### 简介

本文档主要介绍阿里绿网-内容检测API SDK的使用, 在使用SDK前您需要阅读<<内容检测API文档>>, 了解各个接口的具体功能。

## SDK 说明

智能鉴黄、OCR图片识别我们在SDK实现上包装成了一个实现接口，即图片检测SDK接口，分成两种调用方式：

- i. 同步图片检测接口：为了保证性能只支持单张图片调用，将会实时返回图片检测结果
- ii. 异步图片检测接口：支持批量图片进行检测，接口将针对每一张图片返回一个taskId，用户需要根据这个taskId在1分钟后来获取检测结果

图片检测接口同时支持多个场景，如智能鉴黄、OCR图片识别，可以在一次接口调用中针对两个场景进行同时检测，用户只需要在调用接口时传递多个场景参数即可。

## 内容检测API SDK 支持的语言或环境

- Java SDK
- PHP SDK
- Python SDK

## SDK 使用示例程序下载

- green-sdk-sample\_doc

## Java SDK

### 简介

内容检测API SDK包含阿里云Java SDK公共部分和内容检测部分，公共部分依赖 aliyun-java-sdk-core, 内容检测部分依赖aliyun-java-sdk-green

### 安装

您可以使用Maven仓库引入内容检测API SDK或手动下在SDK jar后手动添加到项目中。

### Maven Dependencies

```


```



```

        <dependency>
        <groupId>com.aliyun</groupId>
        <artifactId>aliyun-java-sdk-core</artifactId>
        <optional>true</optional>
        <version>2.4.2</version>
        </dependency>

        <dependency>
        <groupId>com.aliyun</groupId>
        <artifactId>aliyun-java-sdk-green</artifactId>
        <version>1.4.0</version>
        </dependency>
    
```

## 下载jar包

- aliyun-java-sdk-core-2.4.2.jar
- aliyun-java-sdk-green-1.4.0.jar

## 示例

### 异步图片检测接口

```

import com.alibaba.fastjson.JSON;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20160621.ImageDetectionRequest;
import com.aliyuncs.green.model.v20160621.ImageDetectionResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.Arrays;
import java.util.List;

/**
 * 异步图片检测接口, 调用不会实时返回检测结果,
 * 需要根据调用结果返回的taskId在配合"获取图片检测结果接口"使用, 来获取检测结果, 参考
 * AsyncImageDetectionResultSample中的示例
 */
public class AsyncImageDetectionSample extends BaseSample{

    public static void main(String[] args) {
        //请替换成你自己的accessKeyId、accessKeySecret
        IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId, accessKeySecret);
        IAcsClient client = new DefaultAcsClient(profile);

        ImageDetectionRequest imageDetectionRequest = new ImageDetectionRequest();

        /**
         * 设置成异步调用
         * true: 异步
        
```

```

*/
imageDetectionRequest.setAsync(true);

/**设置服务场景
 * 异步图片检测支持多个场景:
 * porn: 黄图检测
 * ocr: ocr文字识别
 * illegal: 暴恐敏感检测
 */

//场景请根据业务自需要设置, 不一定要设置所有的场景
imageDetectionRequest.setScenes(Arrays.asList("porn"));

/**
 * 设置检测的图片链接
 * 异步图片检测一次支持最多50张图片进行检测, 请传递数组
 */
imageDetectionRequest.setImageUrls(Arrays.asList("http://rifleman-share.oss-cn-
hangzhou.aliyuncs.com/test/qrcode/IADOSG0Wo8z2zPk_249_246.jpg"));

try {
ImageDetectionResponse imageDetectionResponse = client.getAcsResponse(imageDetectionRequest);

System.out.println(JSON.toJSONString(imageDetectionResponse));
if("Success".equals(imageDetectionResponse.getCode())){
List<ImageDetectionResponse.ImageResult> imageResults = imageDetectionResponse.getImageResults();
if(imageResults != null && imageResults.size() > 0) {
for (ImageDetectionResponse.ImageResult imageResult : imageResults) {
String taskId = imageResult.getTaskId();
System.out.println(taskId);
/**
 * 异步检测, 请根据taskId, 在1分钟后去调用ImageResultRequest接口, 获取异步图片检测结果
 */
}
}

}else{
/**
 * 检测失败
 */
System.out.println(JSON.toJSONString(imageDetectionResponse));
}

} catch (ServerException e) {
e.printStackTrace();
} catch (ClientException e) {
e.printStackTrace();
}
}
}
}
}

```

## 获取异步图片检测结果接口

```
import com.alibaba.fastjson.JSON;
```

```

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20160621.ImageResultRequest;
import com.aliyuncs.green.model.v20160621.ImageResultResponse;
import com.aliyuncs.green.model.v20160621.ImageResultsRequest;
import com.aliyuncs.green.model.v20160621.ImageResultsResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.Arrays;
import java.util.List;

/**
 * 获取异步图片检测结果接口示例, 包括:
 * 单个异步图片检测结果获取
 * 批量异步图片检测结果获取
 * 注: 获取异步图片检测结果接口依赖于异步图片检测接口返回的taskId, 通过taskId才能获取异步图片检测接口结果, 参照
 * :AsyncImageDetectionSample
 */
public class AsyncImageDetectionResultSample extends BaseSample{

    public static void main(String[] args) {
        //请替换成你自己的accessKeyId、accessKeySecret
        IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId, accessKeySecret);

        IAcsClient client = new DefaultAcsClient(profile);

        //单个异步图片检测结果获取
        //注: taskId 来自 调用异步图片检测时返回的taskId
        String taskId = "fcd8ff5e-0d3a-4e75-bab1-872005e9878f-1468301908864";
        ImageResultRequest imageResultRequest = new ImageResultRequest();
        imageResultRequest.setTaskId(taskId);
        try {
            ImageResultResponse imageResultResponse = client.getAcsResponse(imageResultRequest);
            System.out.println(JSON.toJSONString(imageResultResponse));
            //获取结果成功, 且任务已经处理完成, 做处理
            if("Success".equals(imageResultResponse.getCode())
                && "TaskProcessSuccess".equals(imageResultResponse.getStatus())) {
                ImageResultResponse.ImageResult imageResult = imageResultResponse.getImageResult();

                //porn场景对应的检测结果放在pornResult字段中
                //ocr场景对应的检测结果放在ocrResult字段中
                //illegal场景对应的检测结果放在illegalResult字段中
                //请按需获取
            /**
             * 黄图检测结果
             */
            ImageResultResponse.ImageResult.PornResult pornResult = imageResult.getPornResult();
            if (pornResult != null) {
            /**
             * 绿网给出的建议值, 0表示正常, 1表示色情, 2表示需要review
             */
                Integer label = pornResult.getLabel();
            }
        }
    }
}

```

```

/**
 * 黄图分值, 0-100
 */
Float rate = pornResult.getRate();

// 根据业务情况来做处理
}

//其他的场景的类似向黄图检测结果一样去get结果, 进行处理
imageResult.getOcrResult();
imageResult.getIllegalResult();
}else{
//未处理成功
System.out.println(JSON.toJSONString(imageResultResponse));
}
} catch (ServerException e) {
e.printStackTrace();
} catch (ClientException e) {
e.printStackTrace();
}

//批量异步图片检测结果获取, 最多支持100个taskId
ImageResultsRequest imageResultsRequest = new ImageResultsRequest();
imageResultsRequest.setTaskIds(Arrays.asList("fcd8ff5e-0d3a-4e75-bab1-872005e9878f-1468301908864"));
try {
ImageResultsResponse imageResultsResponse = client.getAcsResponse(imageResultsRequest);

System.out.println(JSON.toJSONString(imageResultsResponse));

if("Success".equals(imageResultsResponse.getCode())) {
List<ImageResultsResponse.ImageDetectResult> imageDetectResults =
imageResultsResponse.getImageDetectResults();

if(imageDetectResults != null && imageDetectResults.size() > 0) {
for (ImageResultsResponse.ImageDetectResult imageDetectResult : imageDetectResults) {
if("Success".equals(imageResultsResponse.getCode())
&& "TaskProcessSuccess".equals(imageDetectResult.getStatus())) {
ImageResultsResponse.ImageDetectResult.ImageResult imageResult = imageDetectResult.getImageResult();

//porn场景对应的检测结果放在pornResult字段中
//ocr场景对应的检测结果放在ocrResult字段中
//illegal场景对应的检测结果放在illegalResult字段中
//请按需获取
/**
 * 黄图检测结果
 */
ImageResultsResponse.ImageDetectResult.ImageResult.PornResult pornResult = imageResult.getPornResult();
if (pornResult != null) {
/**
 * 绿网给出的建议值, 0表示正常, 1表示色情, 2表示需要review
 */
Integer label = pornResult.getLabel();

/**

```

```

* 黄图分值, 0-100
*/
Float rate = pornResult.getRate();

// 根据业务情况来做处理
}

//其他的场景的类似向黄图检测结果一样去get结果, 进行处理
}else{
//未处理成功
System.out.println(JSON.toJSONString(imageDetectResult));
}
}
}
}

} catch (ServerException e) {
e.printStackTrace();
} catch (ClientException e) {
e.printStackTrace();
}

}

}

```

## 同步图片检测接口

```

import com.alibaba.fastjson.JSON;
import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.exceptions.ClientException;
import com.aliyuncs.exceptions.ServerException;
import com.aliyuncs.green.model.v20160621.ImageDetectionRequest;
import com.aliyuncs.green.model.v20160621.ImageDetectionResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;

import java.util.Arrays;
import java.util.List;

/**
 * 同步图片检测结果调用示例, 调用会实时返回检测结果
 */
public class SyncImageDetectionSample extends BaseSample{

    public static void main(String[] args) {
        //请替换成你自己的accessKeyId、accessKeySecret
        IClientProfile profile = DefaultProfile.getProfile("cn-hangzhou", accessKeyId, accessKeySecret);
        IAcsClient client = new DefaultAcsClient(profile);

        ImageDetectionRequest imageDetectionRequest = new ImageDetectionRequest();
    }
}

```

```

/**
 * 是否同步调用
 * false: 同步
 */
imageDetectionRequest.setAsync(false);

/**
 * 同步图片检测支持多个场景:
 * porn: 黄图检测
 * ocr: ocr文字识别
 * illegal: 暴恐敏感检测
 */
//imageDetectionRequest.setScenes(Arrays.asList("porn", "ocr", "illegal"));
imageDetectionRequest.setScenes(Arrays.asList("porn"));

/**
 * 同步图片检测一次只支持单张图片进行检测
 */
imageDetectionRequest.setImageUrls(Arrays.asList("http://rifleman-share.oss-cn-
hangzhou.aliyuncs.com/test/qrcode/IADOSG0Wo8z2zPk_249_246.jpg"));

try {
ImageDetectionResponse imageDetectionResponse = client.getAcsResponse(imageDetectionRequest);

System.out.println(JSON.toJSONString(imageDetectionResponse));
if("Success".equals(imageDetectionResponse.getCode())){
List<ImageDetectionResponse.ImageResult> imageResults = imageDetectionResponse.getImageResults();
if(imageResults != null && imageResults.size() > 0) {
//同步图片检测只有一个返回的ImageResult
ImageDetectionResponse.ImageResult imageResult = imageResults.get(0);

//porn场景对应的检测结果放在pornResult字段中
//ocr场景对应的检测结果放在ocrResult字段中
//illegal场景对应的检测结果放在illegalResult字段中
//请按需获取
/**
 * 黄图检测结果
 */
ImageDetectionResponse.ImageResult.PornResult pornResult = imageResult.getPornResult();
if(pornResult != null) {
/**
 * 绿网给出的建议值, 0表示正常, 1表示色情, 2表示需要review
 */
Integer label = pornResult.getLabel();

/**
 * 黄图分值, 0-100
 */
Float rate = pornResult.getRate();

// 根据业务情况来做处理
}

//其他的场景的类似向黄图检测结果一样去get结果, 进行处理

```

```

imageResult.getOcrResult();
imageResult.getIllegalResult();
imageResult.getSpamResult();
}

}else{
/**
 * 检测失败
 */
}

} catch (ServerException e) {
e.printStackTrace();
} catch (ClientException e) {
e.printStackTrace();
}
}
}
}

```

## Release Notes

Version: 1.4.0 2016年06月21日发布

- 支持阿里云RAM子账号调用,子账号请授权给系统策略:AliyunYundunGreenWebFullAccess

# PHP SDK

## PHP SDK开发包下载

下载地址：

- aliyun-php-sdk-core
- aliyun-php-sdk-green

## 示例

### 异步图片检测接口

```

<?php
/**
 * Date: 16/2/23
 * Time: 17:46
 * 异步图片检测样例
 */

```

```

include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20160621 as Green;

$ak = parse_ini_file("aliyun.ak.ini");
//请替换成你自己的accessKeyId、accessKeySecret
$iClientProfile = DefaultProfile::getProfile("cn-hangzhou", $ak["accessKeyId"], $ak["accessKeySecret"]); // TODO

$client = new DefaultAcsClient($iClientProfile);

// 图片检测
$request = new Green\ImageDetectionRequest ();

//设置参数
$request->setAsync("true");
//设置图片链接, 最多支持50张图片
$request->setImageUrl(json_encode(array("http://rifleman-share.oss-cn-hangzhou.aliyuncs.com/test/qrcode/IADOSG0Wtc0BIs0B-A_504_290.jpg", "http://static.vidmate.mobi/images/movie/e/0/tt4272420/backdrop_w1280.jpg")));
//设置检测的场景
//同步支持多个场景同时识别
//porn: 黄图检测
//ocr : ocr文字识别
//illegal: 暴恐涉政识别
$request->setScene(json_encode(array("porn")));

try {
    $response = $client->getAcsResponse($request);
    print_r($response);

    if("Success" == $response->Code){
        $imageResults = $response->ImageResults->ImageResult;
        foreach ($imageResults as $imageResult) {
            print_r($imageResult->TaskId);
            print_r("\n");
            //保存taskId, 过1分钟来获取对应的内容, 见AsyncImageDetectionResultSample.php
        }
    } catch (Exception $e) {
        print_r($e);
    }
}
    
```

## 获取图片检测结果接口

```

<?php
/**
 * Date: 16/2/23
 * Time: 17:46
 * 异步获取图片检查结果样例,
 * 包括 :
 * 获取单个图片检测结果接口
 * 获取批量图片检测结果接口
    
```



```

*
* 注: 获取异步图片检测结果接口依赖于异步图片检测接口返回的taskId, 通过taskId才能获取异步图片检测接口结果, 参照
AsyncImageDetectionSample.php
*/

include_once 'aliyuncs/aliyun-php-sdk-core/Config.php';
use Green\Request\V20160621 as Green;

$ak = parse_ini_file("aliyun.ak.ini");
//请替换成你自己的accessKeyId、accessKeySecret
$iClientProfile = DefaultProfile::getProfile("cn-hangzhou", $ak["accessKeyId"], $ak["accessKeySecret"]); // TODO

$client = new DefaultAcsClient($iClientProfile);

//获取单个图片检测结果接口
$getTaskResultRequest = new Green\ImageResultRequest();

//设置taskId
$taskId="a72974e3-9d9e-4e1a-81ce-29e227e98a24-1468303410866";
$getTaskResultRequest->setTaskId($taskId);

try {
    $getTaskResultResponse = $client->getAcsResponse($getTaskResultRequest);

    print_r($getTaskResultResponse);

    //任务处理完成在做处理
    if("Success" == $getTaskResultResponse->Code && "TaskProcessSuccess" == $getTaskResultResponse->Status){
        $imageResult = $getTaskResultResponse->ImageResult;

        //黄图结果处理
        if(property_exists($imageResult, "PornResult")) {
            $pornResult = $imageResult->PornResult;
            if(!empty($pornResult)) {
                //打印黄图分值,0-100
                print_r($pornResult->Rate);
                print_r("\n");
                //打印参考建议, 0表示正常, 1表示色情, 2表示需要review
                print_r($pornResult->Label);
                print_r("\n");
            }
        }

        //暴恐涉政结果处理
        if(property_exists($imageResult, "IllegalResult")) {
            $illegalResult = $imageResult->IllegalResult;
            if(!empty($illegalResult)) {
                //打印暴恐涉政分值,0-100
                print_r($illegalResult->Rate);
                print_r("\n");
                //打印参考建议, 0表示正常, 1表示暴恐涉政, 2表示需要review
                print_r($illegalResult->Label);
                print_r("\n");
            }
        }
    }
}
    
```

```

}

//ocr结果处理
//打印ocr结果
if(property_exists($imageResult, "OcrResult")) {
    $ocrResult = $imageResult->OcrResult;
    if(!empty($ocrResult)) {
        print_r($ocrResult->Text);
        print_r("\n");
    }
}

}
} catch (Exception $e) {
    print_r($e);
}

//获取批量图片检测结果接口, 最多支持100个taskId
$getTaskResultsRequest = new Green\ImageResultsRequest();

//json字符串数据格式
$getTaskResultsRequest->setTaskId(json_encode(array("070542bf-10dc-4c21-9c6f-3d228d2de1c0-1457330201321", "a32c9164-577c-418f-903c-1c5b51ed986a-1457330201321")));
try {
    $getTaskResultsResponse = $client->getAcsResponse($getTaskResultsRequest);
    print_r($getTaskResultsResponse);

    if("Success" == $getTaskResultsResponse->Code){
        $imageDetectResults = $getTaskResultsResponse->ImageDetectResults->ImageDetectResult;

        foreach ($imageDetectResults as $imageDetectResult) {
            //任务处理完成在做处理
            if("TaskProcessSuccess" == $imageDetectResult->Status){
                $imageResult = $imageDetectResult->ImageResult;

                //黄图结果处理
                $pornResult = $imageResult->PornResult;
                if(!empty($pornResult)) {
                    //打印黄图分值,0-100
                    print_r($pornResult->Rate);
                    print_r("\n");
                    //打印参考建议, 0表示正常, 1表示色情, 2表示需要review
                    print_r($pornResult->Label);
                    print_r("\n");
                }

                //暴恐涉政结果处理
                $illegalResult = $imageResult->IllegalResult;
                if(!empty($illegalResult)) {
                    //打印暴恐涉政分值,0-100
                    print_r($illegalResult->Rate);
                    print_r("\n");
                }
            }
        }
    }
}

```

```

//打印参考建议, 0表示正常, 1表示暴恐涉政, 2表示需要review
print_r($illegalResult->Label);
print_r("\n");
}

//ocr结果处理
//打印ocr文本结果
$ocrResult = $imageResult->OcrResult;
if(!empty($ocrResult)){
print_r($ocrResult->Text);
print_r("\n");
}
}
}

} catch (Exception $e) {
print_r($e);
}
    
```

## 同步图片检测接口

```

<?php
/**
 * Date: 16/2/23
 * Time: 17:46
 * 同步图片检测样例, 调用会实时返回检测结果
 */

include_once 'aliyun/aliyun-php-sdk-core/Config.php';
use Green\Request\V20160621 as Green;

$ak = parse_ini_file("aliyun.ak.ini");
//请替换成你自己的accessKeyId、accessKeySecret
$iClientProfile = DefaultProfile::getProfile("cn-hangzhou", $ak["accessKeyId"], $ak["accessKeySecret"]); // TODO

$client = new DefaultAcsClient($iClientProfile);

// 图片检测
$request = new Green\ImageDetectionRequest ();

//设置参数
//设置为同步调用
$request->setAsync("false");
//设置图片链接
//同步只支持单张图片
$request->setImageUrl(json_encode(array("http://rifleman-share.oss-cn-hangzhou.aliyuncs.com/test/qrcode/IADOSG0Wtc0BIs0B-A_504_290.jpg")));
//设置检测的场景
//同步支持多个场景同时识别
    
```

```

//porn: 黄图检测
//ocr : ocr文字识别
//illegal: 暴恐涉政识别
//场景请根据自己业务来选择, 同步调用一次不建议过多个场景, 可能超时概率会很高
$request->setScene(json_encode(array("porn")));

try {
$response = $client->getAcsResponse($request);
print_r($response);

//返回状态值成功时进行处理
if("Success" == $response->Code){
$imageResults = $response->ImageResults->ImageResult;
foreach ($imageResults as $imageResult) {
//黄图结果处理
$pornResult = $imageResult->PornResult;
if(!empty($pornResult)) {
//打印黄图分值,0-100
print_r($pornResult->Rate);
print_r("\n");
//打印参考建议, 0表示正常, 1表示色情, 2表示需要review
print_r($pornResult->Label);
print_r("\n");
}

//暴恐涉政结果处理
$illegalResult = $imageResult->IllegalResult;
if(!empty($illegalResult)) {
//打印暴恐涉政分值,0-100
print_r($illegalResult->Rate);
print_r("\n");
//打印参考建议, 0表示正常, 1表示暴恐涉政, 2表示需要review
print_r($illegalResult->Label);
print_r("\n");
}

//ocr结果处理
//打印ocr结果
if(!empty($ocrResults)) {
$ocrResult = $imageResult->OcrResult;
print_r($ocrResult->Text);
print_r("\n");
}

}
} catch (Exception $e) {
print_r($e);
}
    
```

# Python SDK

## 通过pip安装

```
sudo pip install aliyun-python-sdk-core
sudo pip install aliyun-python-sdk-green
```

## 示例

### 异步图片检测接口

```
#coding=utf-8

# 异步调用图片服务接口,调用不会实时返回检测结果,
# 需要根据调用结果返回的taskid在配合"获取图片检测结果接口"使用,来获取检测结果,参考
# AsyncImageDetectionResultSample中的示例

from aliynsdckore import client
from aliynsdckgreen.request.v20160621 import ImageDetectionRequest
import json

import ConfigParser
cf = ConfigParser.ConfigParser()
cf.read("aliyun.ak.conf")
# 请替换成你自己的accessKeyId、accessKeySecret,您可以类似的配置在配置文件里面,也可以直接明文替换
clt = client.AcsClient(cf.get("AK", "accessKeyId"), cf.get("AK", "accessKeySecret"), 'cn-hangzhou')

request = ImageDetectionRequest.ImageDetectionRequest()
request.set_accept_format('json')

# 设置为异步调用
request.set_Async('true')
# 设置要检测的图片链接
# 异步支持多张图片,最多50张,请按需传递, json字符串格式
request.set_ImageUrl(json.dumps(["http://rifleman-share.oss-cn-hangzhou.aliyuncs.com/test/qrcode/IADOSG0Wtc0BIs0B-A_504_290.jpg"]))
# 设置要检测的服务场景
# 异步支持多个场景同时识别
# porn: 黄图检测
# ocr : ocr文字识别
# illegal: 暴恐涉政识别
request.set_Scene(json.dumps(["porn", "ocr", "illegal"]))
request.set_Scene(json.dumps(["porn"]))
response = clt.do_action(request)

# 返回结果格式参照response/AsyncImageDetectionResponse.json
print response
```

```

result = json.loads(response)

if("Success" == result["Code"]) :
    imageResults = result["ImageResults"]["ImageResult"];
    #一张图片对应一个taskId, 如果是多张图片检测, 请自行遍历
    for imageDetectResult in imageResults:
        # 根据taskId获取异步图片检测结果, 参照AsyncImageDetectionResultSample.py
        taskId = imageDetectResult["TaskId"]
        print taskId
    else:
        print result
    
```

## 获取异步图片检测结果接口

```

#coding=utf-8

# 获取异步图片检测结果的接口
# 包括:
# 获取单个异步图片检测结果接口
# 获取批量异步图片检测结果接口
# 注: 获取异步图片检测结果接口依赖于异步图片检测接口返回的taskId, 通过taskId才能获取异步图片检测接口结果
AsyncImageDetectionSample.py

from aliynsdkcore import client
from aliynsdkgreen.request.v20160621 import ImageResultRequest
from aliynsdkgreen.request.v20160621 import ImageResultsRequest
import json

import ConfigParser
cf = ConfigParser.ConfigParser()
cf.read("aliyun.ak.conf")
# 请替换成你自己的accessKeyId、accessKeySecret, 您可以类似的配置在配置文件里面, 也可以直接明文替换
clt = client.AcsClient(cf.get("AK", "accessKeyId"), cf.get("AK", "accessKeySecret"), 'cn-hangzhou')

# 获取单个异步图片检测结果接口
request = ImageResultRequest.ImageResultRequest()
request.set_accept_format('json')

# 设置taskId
request.set_TaskId('25481d60-9987-4d5f-9a7c-4c1c06748230-1468304490867');
response = clt.do_action(request)

print response
result = json.loads(response)

# 任务处理完成在做处理
if "Success" == result["Code"] and "TaskProcessSuccess" == result["Status"]:
    imageResult = result["ImageResult"]
    print imageResult
    # 黄图检测结果
    pornResult = imageResult.get("PornResult")
    if not pornResult is None:
        print pornResult
    # 黄图结果中, 包含检测分值, 和参考建议
    
```

```

# 打印检测分值,0-100
print pornResult.get("Rate")
# 打印参考建议, 0表示正常, 1表示色情, 2表示需要review
print pornResult.get("Label")

# ocr结果
# 返回检测出来的文本数组
ocrResult = imageResult.get("OcrResult")
if not ocrResult is None:
    print ocrResult.get("Text")

# 暴恐涉政结果
illegalResult = imageResult.get("IllegalResult")
if not illegalResult is None:
    # 分值:0-100
    print illegalResult.get("Rate")
    # 参考建议: 0表示正常, 1表示暴恐涉政, 2表示需要review
    print illegalResult.get("Label")

else :
    print response

# 获取多个异步图片检测结果接口, 最多支持100个taskId
request = ImageResultsRequest.ImageResultsRequest()
request.set_accept_format('json')

# 设置taskId
request.set_TaskId(json.dumps(['25481d60-9987-4d5f-9a7c-4c1c06748230-1468304490867']));
response = clt.do_action(request)
print response
result = json.loads(response)

if "Success" == result["Code"]:
    imageDetectResults = result["ImageDetectResults"]["ImageDetectResult"]
    for imageDetectResult in imageDetectResults:
        if "TaskProcessSuccess" == imageDetectResult["Status"]:
            imageResult = imageDetectResult["ImageResult"]
            print imageResult
            # 黄图检测结果
            pornResult = imageResult.get("PornResult")
            if not pornResult is None:
                print pornResult
                # 黄图结果中, 包含检测分值, 和参考建议
                # 打印检测分值,0-100
                print pornResult.get("Rate")
                # 打印参考建议, 0表示正常, 1表示色情, 2表示需要review
                print pornResult.get("Label")

            # ocr结果
            # 返回检测出来的文本数组
            ocrResult = imageResult.get("OcrResult")
            if not ocrResult is None:
                print ocrResult.get("Text")

            # 暴恐涉政结果
    
```

```

illegalResult = imageResult.get("IllegalResult")
if not illegalResult is None:
# 分值:0-100
print illegalResult.get("Rate")
# 参考建议: 0表示正常, 1表示暴恐涉政, 2表示需要review
print illegalResult.get("Label")

print result
    
```

## 同步图片检测接口

```

#coding=utf-8

# 同步图片检测服务接口, 会实时返回检测的结果

from aliyunSdkcore import client
from aliyunSdkgreen.request.v20160621 import ImageDetectionRequest
import json

import ConfigParser
cf = ConfigParser.ConfigParser()
cf.read("aliyun.ak.conf")
# 请替换成你自己的accessKeyId、accessKeySecret, 您可以类似的配置在配置文件里面, 也可以直接明文替换
clt = client.AcsClient(cf.get("AK", "accessKeyId"), cf.get("AK", "accessKeySecret"), 'cn-hangzhou')

request = ImageDetectionRequest.ImageDetectionRequest()
request.set_accept_format('json')

# 设置成同步调用
request.set_Async('false')
# 设置要检测的图片链接
# json字符串格式, 同步只支持单张图片
request.set_ImageUrl(json.dumps(["http://rifleman-share.oss-cn-hangzhou.aliyuncs.com/test/qrcode/IADOSG0Wtc0BIs0B-A_504_290.jpg"]))
# 设置要检测的服务场景
# 异步支持多个场景同时识别
# porn: 黄图检测
# ocr : ocr文字识别
# illegal: 暴恐涉政识别
request.set_Scene(json.dumps(["porn"]))
response = clt.do_action(request)

print response

result = json.loads(response)

if "Success" == result["Code"]:
imageResults = result["ImageResults"]["ImageResult"]
for imageDetectResult in imageResults:
print imageDetectResult
# 黄图检测结果
pornResult = imageDetectResult["PornResult"]
print pornResult
# 黄图结果中, 包含检测分值, 和参考建议
    
```



```

# 打印检测分值,0-100
print pornResult.get("Rate")
# 打印参考建议, 0表示正常, 1表示色情, 2表示需要review
print pornResult.get("Label")

# ocr结果
# 返回检测出来的文本数组
ocrResult = imageDetectResult["OcrResult"]
print ocrResult.get("Text")

# 暴恐涉政结果
illegalResult = imageDetectResult["IllegalResult"]
# 分值:0-100
print illegalResult.get("Rate")
# 参考建议: 0表示正常, 1表示暴恐涉政, 2表示需要review
print illegalResult.get("Label")
    
```

## 附录

### 内容检测API返回结果状态码

code	msg	说明
Success	调用成功	调用绿网开放接口成功
InvalidArgument	调用参数无效: %s	传入参数有误
SceneNull	算法场景不能为空	传入算法场景为空
UrlOrFile	有且只能有image_url或image_file的一种, 不能同时使用	设置image_url值, 或设置image_file值, 不能同时设置
Failed	调用失败	调用绿网开放接口失败
ServerInnerException	服务端错误(%s), 请稍后重试	绿网内部出现异常
TaskResultNotFound	未查找到指定结果	出入的taskid不存在
AlgorithmTimeOut	图片/文本算法检测超时	后端算法计算超时
ImageTooLarge	图片尺寸过大	单张图片尺寸超出5M
ProcessImageContentFailed	图片内容处理失败, 请稍后重试	后端算法出现异常
ImageDetectFailed	图片检测失败	图片格式可能不支持
UrlAndFileEmpty	连接和文件都为空	设置image_url值, 或设置image_file值, 不能都不设置