

# 数据风控

## 用户指南

# 用户指南

## 人机验证-前端代码集成

## 智能验证

## H5版本

## 简介

智能验证以一个button的形式提供给接入方集成人机识别的能力，大部分用户只需要一次点击交互即可完成原先复杂的人机验证码操作



点击开始验证

## 完整接入demo

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<script>
window.NVC_Opt = {
  appkey:'CF_APP_1',
  scene:'nvc_register',
```

```

renderTo:'#captcha',
trans: {"key1": "code0", "nvcCode":200},
elements: [
  '//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png',
  '//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png'
],
bg_back_prepared: '//img.alicdn.com/tps/TB1skE5SFXXXb3XXXXXXXXXXXX-100-80.png',
bg_front:
'data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAAGQAAABQCAMAAADY1yDdAAAAABGdBTUEAALGPC/xhB
QAAAFzUkdCAK7OHOkAAAADUExURefk5w+ruswAAAAFsURBVfjD7cExAQAAAMKg9U9tCU+gAAAAAAAAAIC3AR
+QAAFPIUGoAAAAAEIFTkSuQmCC',
obj_ok: '//img.alicdn.com/tfs/TB1rmyTltfJ8KJy0FeXXXKEXXa-50-74.png',
bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCSVXXXAsFXfXXXXXXXXXXXX-100-80.png',
obj_error: '//img.alicdn.com/tfs/TB1q9yTltfJ8KJy0FeXXXKEXXa-50-74.png',
bg_back_fail: '//img.alicdn.com/tfs/TB1w2oOSFXXXb4XpXXXXXXXXXXXX-100-80.png',
upLang:{"cn":{"
  _ggk_guide: "请在屏幕上滑动，刮出两面盾牌",
  _ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",
  _ggk_loading: "加载中",
  _ggk_fail: ['呀，盾牌不见了<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
  _ggk_action_timeout: ['我等得太久啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
  _ggk_net_err: ['网络实在不给力<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
  _ggk_too_fast: ['您刮得太快啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题']
  }
  }
  }
</script>

<script src="//g.alicdn.com/sd/nvc/1.1.112/guide.js"></script>
<script>
window.onload = function(){
var ic = new smartCaptcha({
renderTo: '#sc',
width: 300,
height: 42,
default_txt: '点击按钮开始智能验证',
success_txt: '验证成功',
fail_txt: '验证失败，请在此点击按钮刷新',
scanning_txt: '智能检测中',
success: function(data) {
console.log(NVC_Opt.token)
console.log(data.sessionID);
console.log(data.sig);
},
});
ic.init();
}

</script>

</head>
<body>

```

```
<script src="//g.alicdn.com/sd/smartCaptcha/0.0.1/index.js"></script>
<div id="sc">
</div>
</body>
</html>
```

## 接入注意事项

1.window.NVC\_Opt全局变量的声明一定要在“ //g.alicdn.com/sd/nvc/1.1.112/guide.js” 资源引入之前配置完毕

2.new smartCaptcha的声明需要在window.NVC\_Opt和“ //g.alicdn.com/sd/nvc/1.1.112/guide.js” 资源引入之后配置

## 参数详细说明

### window.NVC\_Opt

参数	含义
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
trans	业务键字段。您可以不写此字段，也可以按照下文中“ 问题排查” 部分文档配置此字段，便于线上问题排查
elements	两个长度的List，自定义需要刮出的两个elements的图片资源
bg_back_prepared	自定义刮动时的背景图像，平铺填充
bg_front	自定义刮动的前景（浮层），注意这里只接受base64数据流
obj_ok	自定义验证通过时显示的图标资源
bg_back_pass	自定义验证通过时的背景图像，平铺填充
obj_error	自定义验证失败或异常时显示的图标资源
bg_back_fail	自定义验证失败或异常时显示的背景图像，平铺填充
upLang	用于自定义刮刮卡文案，具体可以参见“ 刮刮卡交互验证前端代码集成文档”

## new smartCaptcha配置项

参数	含义
renderTo	配置智能验证按钮渲染的位置
width	按钮长度，最小值300，默认300
height	按钮高度，最小值42，默认42
default_txt	配置按钮等待交互状态的文案
success_txt	配置验证通过的文案
fail_txt	配置验证拦截的文案
scaning_txt	配置检测状态中的文案
success	验证成功的回调，请在此回调中保留token、sessionId和sig参数用作验签

## 测试指南

### (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or二次验证or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
{ "nvcCode" :200}	智能验证直接通过
{ "nvcCode" :400, "key1" : "code0" }	智能验证会唤醒滑动验证，且滑动验证一定通过
{ "nvcCode" :400, "key1" : "code300" }	智能验证会唤醒滑动验证，且滑动验证一定拦截
{ "nvcCode" :600, "key1" : "code0" }	智能验证会唤醒刮刮卡验证，且刮刮卡验证一定通过
{ "nvcCode" :600, "key1" : "code300" }	智能验证会唤醒刮刮卡验证，且刮刮卡验证一定拦截
{ "nvcCode" :800}	智能验证直接拦截

### (2) 正式上线测试

正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：

·请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写

·请注意您当前验证码的使用场景。如果您的验证码在PCweb场景下使用，请不要用手机等移动设备进行测试  
·在测试低版本IE浏览器时，不要用IE11浏览器的兼容模式去模拟生成低版本浏览器。请用真正低版本IE浏览器进行测试

## PC-WEB版本

### 简介

智能验证以一个button的形式提供给接入方集成人机识别的能力，大部分用户只需要一次点击交互即可完成原先复杂的人机验证码操作



### 完整接入demo

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<script>
window.NVC_Opt = {
  appkey:'CF_APP_1',
  scene:'nvc_register',
  renderTo:'#captcha',
  trans: {"key1": "code0", "nvcCode":200},
  elements: [
    '//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png',
    '//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png'
  ],
  bg_back_prepared: '//img.alicdn.com/tps/TB1skE5SFXXXb3XXXXXXXXXXXX-100-80.png',
  bg_front:
'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAGQAAABQCAMAAADY1yDdAAAAABGdBTUEAALGPC/xhB
QAAAAFzUkdCAK7OHOkAAADUEXURfk5w+ruswAAAAFSURBVfjD7cExAQAAAMKg9U9tCU+gAAAAAAAAAIC3AR
+QAAFPIUGoAAAAAEIFTkSuQmCC',
  obj_ok: '//img.alicdn.com/tfs/TB1rmyTltfJ8KJy0FeXXXKEXXa-50-74.png',
  bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCSVXXXasXFXXXXXXXXXXXX-100-80.png',
  obj_error: '//img.alicdn.com/tfs/TB1q9yTltfJ8KJy0FeXXXKEXXa-50-74.png',
  bg_back_fail: '//img.alicdn.com/tfs/TB1w2oOSFXXXb4XpXXXXXXXXXXXX-100-80.png',
  upLang:{"cn":{"
```

```
_ggk_guide: "请摁住鼠标左键，刮出两面盾牌",
_ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",
_ggk_loading: "加载中",
_ggk_fail: ['呀，盾牌不见了<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_action_timeout: ['我等得太久啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_net_err: ['网络实在不给力<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_too_fast: ['您刮得太快啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题']
}
}
}
</script>

<script src="//g.alicdn.com/sd/nvc/1.1.112/guide.js"></script>
<script>
window.onload = function(){
var ic = new smartCaptcha({
renderTo: '#sc',
width: 300,
height: 42,
default_txt: '点击按钮开始智能验证',
success_txt: '验证成功',
fail_txt: '验证失败，请在此点击按钮刷新',
scanning_txt: '智能检测中',
success: function(data) {
console.log(NVC_Opt.token)
console.log(data.sessionId);
console.log(data.sig);
},
});
ic.init();
}

</script>

</head>
<body>
<script src="//g.alicdn.com/sd/smartCaptcha/0.0.1/index.js"></script>
<div id="sc">
</div>
</body>
</html>
```

## 接入注意事项

- 1.window.NVC\_Opt全局变量的声明一定要在“ //g.alicdn.com/sd/nvc/1.1.112/guide.js” 资源引入之前配置完毕
- 2.new smartCaptcha的声明需要在window.NVC\_Opt和“ //g.alicdn.com/sd/nvc/1.1.112/guide.js” 资源引入之后配置

# 参数详细说明

## window.NVC\_Opt

参数	含义
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
trans	业务键字段。您可以不写此字段，也可以按照下文中“问题排查”部分文档配置此字段，便于线上问题排查
elements	两个长度的List，自定义需要刮出的两个elements的图片资源
bg_back_prepared	自定义刮动时的背景图像，平铺填充
bg_front	自定义刮动的前景（浮层），注意这里只接受base64数据流
obj_ok	自定义验证通过时显示的图标资源
bg_back_pass	自定义验证通过时的背景图像，平铺填充
obj_error	自定义验证失败或异常时显示的图标资源
bg_back_fail	自定义验证失败或异常时显示的背景图像，平铺填充
upLang	用于自定义刮刮卡文案，具体可以参见“刮刮卡交互验证前端代码集成文档”

## new smartCaptcha配置项

参数	含义
renderTo	配置智能验证按钮渲染的位置
width	按钮长度，最小值300，默认300
height	按钮高度，最小值42，默认42
default_txt	配置按钮等待交互状态的文案
success_txt	配置验证通过的文案
fail_txt	配置验证拦截的文案
scaning_txt	配置检测状态中的文案
success	验证成功的回调，请在此回调中保留token、sessionId和sig参数用作验签



# 测试指南

## (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or二次验证or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
{ "nvcCode" :200}	智能验证直接通过
{ "nvcCode" :400, "key1" : "code0" }	智能验证会唤醒滑动验证，且滑动验证一定通过
{ "nvcCode" :400, "key1" : "code300" }	智能验证会唤醒滑动验证，且滑动验证一定拦截
{ "nvcCode" :600, "key1" : "code0" }	智能验证会唤醒刮刮卡验证，且刮刮卡验证一定通过
{ "nvcCode" :600, "key1" : "code300" }	智能验证会唤醒刮刮卡验证，且刮刮卡验证一定拦截
{ "nvcCode" :800}	智能验证直接拦截

## (2) 正式上线测试

正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：

- 请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写
- 请注意您当前验证码的使用场景。如果您的验证码在PCweb场景下使用，请不要用手机等移动设备进行测试
- 在测试低版本IE浏览器时，不要用IE11浏览器的兼容模式去模拟生成低版本浏览器。请用真正低版本IE浏览器进行测试

# 刮刮卡

# H5版本

# 简介

刮刮卡交互验证码(Scratchcard-Captcha)是阿里集团推出的第二代行为验证码产品，它需要用户与页面上刮刮卡UI组件进行交互，刮出特定内容即可通过验证。



# 资源引入说明

使用H5版刮刮卡交互验证码，需要引入以下js资源：

```
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/nch5/index.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8"
src="//aeis.alicdn.com/sd/nch5/index.js?t=2015052012"></script> -->
```

其中，t字段建议设置为小时级别时间戳，随时间更新，确保js不会被浏览器长时间缓存，这样能够享受到我们针对js的动态发布迭代。

# 完整接入demo

```
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0,
user-scalable=0">
<script charset="utf-8" src="//g.alicdn.com/sd/nch5/index.js"></script>
</head>
```

```

<body>
<div id="your-dom-id"></div>
<script>
var nc_token = ["CF_APP_1", (new Date()).getTime(), Math.random()].join(':');
var NC_Opt = {
  renderTo: "#your-dom-id",
  appkey: "CF_APP_1",
  scene: 'test',
  token: nc_token,
  trans: {'key1': "code0"},
  is_Opt: 0,
  type: "scrape",
  width: 300,
  height: 100,
  isEnabled: true,
  language: 'cn',
  times: 5,
  objects: ["//img.alicdn.com/tps/TB1BT9jPFXXXXbyXFXXXXXXXX-80-80.png"], //勿动，照抄即可
  apimap: {
    // 'uab_Url': '//aeu.alicdn.com/js/uac/909.js',
  },
  elements: [
    '//img.alicdn.com/tfs/TB17cwllsLJ8KJjy0FnXXcFDpXa-50-74.png',
    '//img.alicdn.com/tfs/TB17cwllsLJ8KJjy0FnXXcFDpXa-50-74.png'
  ],
  bg_back_prepared: '//img.alicdn.com/tps/TB1skE5SFXXXXb3XXXXXXXXXXXX-100-80.png',
  bg_front:
    'data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAAGQAAABQCAMAAADY1yDdAAAAABGdBtUEAALGPC/xhBQAAAFzUkdCAK7OHOkAAADUExURefk5w+ruswAAAAFSURBVfJD7cExAQAAAMKg9U9tCU+gAAAAAAAAAIC3AR+QAAFPIUGoAAAAAEIFTkSuQmCC',
  obj_ok: '//img.alicdn.com/tfs/TB1rmyTltfJ8KJjy0FeXXXKEXXa-50-74.png',
  bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCSVXXXXXasXFXXXXXXXX-100-80.png',
  obj_error: '//img.alicdn.com/tfs/TB1q9yTltfJ8KJjy0FeXXXKEXXa-50-74.png',
  bg_back_fail: '//img.alicdn.com/tfs/TB1w2oOSFXXXXb4XpXXXXXXXXXX-100-80.png',
  upLang:{"cn":{
    _gk_guide: "请在屏幕上滑动，刮出两面盾牌",
    _gk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",
    _gk_loading: "加载中",
    _gk_fail: ["呀，盾牌不见了<br/>请", "javascript:NoCaptcha.reset()", '再来一次', '或',
      "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
    _gk_action_timeout: ["我等得太久啦<br/>请", "javascript:NoCaptcha.reset()", '再来一次', '或',
      "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
    _gk_net_err: ["网络实在不给力<br/>请", "javascript:NoCaptcha.reset()", '再来一次', '或',
      "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
    _gk_too_fast: ["您刮得太快啦<br/>请", "javascript:NoCaptcha.reset()", '再来一次', '或',
      "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题']
  }
  },
  callback: function(data) { //成功回调
    window.console && console.log(nc_token)
    window.console && console.log(data.sessionId)
    window.console && console.log(data.sig)
  },
  failCallback: function(data) { //拦截or失败回调
  },
  error: function(data) { //异常回调

```

```

}
};
var nc = NoCaptcha.init(NC_Opt);
nc.setEnabled(true); // 启动
</script>
</body>
</html>

```

## 参数详细说明

参数	含义
renderTo	声明刮刮卡需要渲染的目标元素ID
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
type	请写死" scrape"
token	刮刮卡交互验证码的主键，请务必不需要写死固定值。请确保每个用户每次打开页面时，token都是不同的。建议格式为" 您的appkey" + " 时间戳" + " 随机数"
width	刮刮卡长度，最小值300，默认值300
height	刮刮卡宽度，最小值100，默认值100
trans	业务键字段。您可以不写此字段，也可以按照下文中" 问题排查与错误码" 部分文档配置此字段，便于线上问题排查
elementID	通过Dom的ID属性自动填写trans业务键，您可以不写此字段，也可以按照下文中" 问题排查与错误码" 部分文档配置此字段，便于线上问题排查
is_Opt	是否自己配置底层采集组件。如无特殊场景请保持写0或不写此项。默认为0
isEnabled	是否启用，无特殊场景请默认写true。默认为true
language	语言。详细配置方法请见下方" 自定义文案" 部分。默认为cn(中文)
times	允许服务器超时重复次数，默认5次
objects	特殊字段，请务必写死 [ "///img.alicdn.com/tps/TB1BT9jPFXXXXbyXFXXXXXXXXXX-80-80.png" ]
apimap	用于自己指定滑动验证各项请求的接口地址。如无特殊情况，请不要配置这里的apimap
elements	两个长度的List，自定义需要刮出的两个

	elements的图片资源
bg_back_prepared	自定义刮动时的背景图像，平铺填充
bg_front	自定义刮动的前景（浮层），注意这里只接受base64数据流
obj_ok	自定义验证通过时显示的图标资源
bg_back_pass	自定义验证通过时的背景图像，平铺填充
obj_error	自定义验证失败或异常时显示的图标资源
bg_back_fail	自定义验证失败或异常时显示的背景图像，平铺填充
upLang	用于自定义文案，具体参见下方“自定义文案”部分文档
callback	验证成功时会触发此回调
failCallback	拦截或验证失败时会触发此回调
error	异常回调

## 方法详细说明

noCaptcha对象中维护了一些方法提供给您使用，您可以通过诸如`var nc = new noCaptcha(NC_Opt);`  
`nc.functionName(parm);`的方式来调用使用它们。

方法名	参数	作用
reset	无	任何时候调用此方法，可以将刮刮卡重置为初始状态(等待刮动)
hide	无	任何时候调用此方法，可以隐藏刮刮卡组件
show	无	在调用hide方法后，任何时候调用此方法，可以展示刮刮卡组件

## 测试指南

### (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
{ "key1" : " code0" }	刮动一定会通过
{ "key1" : " code300" }	刮动一定会被拦截

## (2) 正式上线测试

正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：

- 请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写
- 请注意您当前验证码的使用场景。如果您的验证码在Html5场景下使用，请不要用PC浏览器进行测试

## 自定义文案

您可以通过配置刮刮卡配置项中的upLang参数来自定义您需要的文案。

```
upLang:{"cn":{
  _ggk_guide: "请摁住鼠标左键，刮出两面盾牌",//初始化完成
  _ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",//验证成功
  _ggk_loading: "加载中",//加载过程
  _ggk_fail: ["呀，盾牌不见了<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//验证失败
  _ggk_action_timeout: ["我等得太久啦<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//操作超时
  _ggk_net_err: ["网络实在不给力<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//加载失败
  _ggk_too_fast: ["您刮得太快啦<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//刮动过快
  }
}
```

对于默认的“盾牌”样式的交互与UI，我们为您准备好了18国翻译，您可以根据自己的需求进行替换。替换方法：将language参数和upLang的第一个参数替换为您想要的语言，并在第二个参数中自定义默认UI(盾牌)的文案(“加载中”、“网络异常”、“刮动过快”、“等的太久”在多语言场景不需要自定义文案，修改language参数后会自动提供翻译)

ar\_SA:阿拉伯文

de\_DE:德文

es\_ES:西班牙文

fr\_FR:法文

in\_ID:印尼语

it\_IT:意大利文

iw\_HE:希伯来语

ja\_JP:日文

ko\_KR:韩文

nl\_NL:荷蘭文  
 pt\_BR:波蘭文  
 ru\_RU:俄文  
 th\_TH:泰文  
 tr\_TR:土耳其文  
 vi\_VN:越南文  
 cn:简体中文  
 tw:繁体中文  
 en:英文

中文	请在屏幕上滑动，刮出两面盾牌	呀，盾牌不见了，请再来一次或反馈问题	恭喜您成功刮出盾牌，继续下一步操作吧
台湾繁体	請在螢幕上滑動，刮出兩個盾牌。	哎呀，盾牌消失了！請再來一次或通報錯誤。	恭喜！您刮出盾牌。請繼續下一步吧！
English	Swipe the screen to scrape out two shields.	Uh oh, the shield disappeared! Please try again or report an error.	Congratulations! You gained a shield. Continue to the next step!
Spanish	Desliza la pantalla para raspar dos escudos.	¡Vaya! El escudo ha desaparecido. Vuelve a intentarlo o informa del problema.	¡Enhorabuena! Has conseguido un escudo. Continúa con el siguiente paso.
Polish	Przesuń palcem po ekranie, aby zdrapać dwie tarcze.	Ojej, tarcza znikła! Spróbuj ponownie lub zgłoś problem.	Gratulacje! Zdobywasz tarczę. Przejdź do kolejnego kroku!
French	Balayez l' écran pour gratter deux écus.	Oups, l' écu a disparu ! Veuillez réessayer ou signaler une erreur.	Félicitations ! Vous avez reçu un écu. Passez à l' étape suivante !
German	Wischen Sie auf dem Bildschirm, um zwei Schilde zu kratzen.	Hoppla, der Schild ist verschwunden! Versuchen Sie es erneut oder melden Sie einen Fehler.	Glückwunsch! Sie haben einen Schild gewonnen. Gehen Sie weiter zum nächsten Schritt!
Italian	Scorrere sullo schermo per grattare due scudi.	Oh oh, lo scudo è scomparso! Riprovare o segnalare un errore.	Congratulazioni! Hai guadagnato uno scudo. Continua con il passaggio successivo.
Russian	Проведите по экрану, чтобы стереть два щита.	Ох, щит пропал! Повторите попытку или сообщите об ошибке.	Поздравляем! Вы получили щит. Перейдите на следующий шаг!
Japanese	画面をスワイプして、2つのシールドを掘り出してください。	シールドが消えました！再度お試しになるか、エラーを報告してください。	おめでとうございます！シールドを獲得しました。次のステップに進んでください！

Korean	.	! .	! . !
Arabic	.	! .	! . !
Turkish	İki kalkanı kazımak için ekranı kaydırın.	Eyvah, kalkan kayboldu! Lütfen daha sonra tekrar deneyin veya bir hata bildiriminde bulunun.	Tebrikler! Bir kalkan kazandınız. Bir sonraki adıma geçin!
Thai		!	! !
Vietnamese	Vut màn hình đ kéo ra hai khiên.	, khiên đã bin mt! Vui lòng th li hoc báo cáo li.	Xin chúc mng! Bn đã giành đc mt khiên. Tip tc bc tip theo!
Dutch	Veeg over het scherm om twee schilden weg te strepen.	Oh, het schild is verdwenen! Probeer het opnieuw of rapporteer uw probleem.	Gefeliciteerd! U hebt een schild gekregen. Ga verder met de volgende stap!
Hebrew	.	-, ! .	! . !
Indonesian	Gesek layar untuk mengkurat dua buah perisai.	Oh, perisainya menghilang! Harap coba lagi atau laporkan kesalahan.	Selamat! Anda mendapatkan sebuah perisai. Lanjutkan ke langkah berikutnya!

## 问题排查

### (1) 上传业务键

刮刮卡交互验证默认不会采集您当前业务场景下的用户名、手机号等业务键，为了方便问题排查，这里强烈建议您配置上传业务主键，这样遇到问题的客户可以提供“业务键+时间”，很方便的关联到对应刮刮卡的请求。否则您必须提供引导用户进行抓包才能具体对应查到日志和请求。

您可以配置elementID来让滑动验证主动采集。elementID的值为一个stringList，配置之后在刮动结束的时刻，滑块会以elementID当中的每一个string作为ID，去获取对应dom的value值并上传。举个例子，如果您配置elementID:[ “usernameID” ]，则刮动结束时document.getElementById( “usernameID” ).value会自动被上传至验证码服务端。

### (2) 获取token

如果您不方便上传业务键，在用户遇到问题时您可以引导其点击“反馈问题”链接，然后将打开页面的url复制保存下来以方便后续查询日志，无需您自己进行抓包。



# PC-WEB版本

## 简介

刮刮卡交互验证码(Scratchcard-Captcha)是阿里集团推出的第二代行为验证码产品，它需要用户与页面上刮刮卡UI组件进行交互，刮出特定内容即可通过验证。



## 资源引入说明

使用PC版刮刮卡交互验证码，需要引入以下js资源：

```
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8" src="//aeis.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
-->
```

其中，t字段建议设置为小时级别时间戳，随时间更新，确保js不会被浏览器长时间缓存，这样能够享受到我们针对js的动态发布迭代

## 完整接入demo

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8" src="//aeis.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
-->
</head>
<body>
<div id="your-dom-id"></div>
<script type="text/javascript">
var nc_token = ["CF_APP_1", (new Date()).getTime(), Math.random()].join(':');
var NC_Opt = {
  renderTo: "#your-dom-id",
  appkey: "CF_APP_1",
  scene: 'test',
  token: nc_token,
  trans: {'key1': "code0"},
  is_Opt: 0,
  type: "scrape",
  width: 300,
  height: 100,
  isEnabled: true,
  language: 'cn',
  times: 5,
  objects: ["//img.alicdn.com/tps/TB1BT9jPFXXXXXbyXFXXXXXXXXXX-80-80.png"],//勿动，照抄即可
  apimap: {
    // 'uab_Url': '//aeu.alicdn.com/js/uac/909.js',
  },
  elements: [
    '//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png',
    '//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png'
  ],
  bg_back_prepared: '//img.alicdn.com/tps/TB1skE5SFXXXXb3XXXXXXXXXXXX-100-80.png',
  bg_front:
'data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAAGQAAABQCAMAAADY1yDdAAAAABGdBtUEAALGPC/xhB
QAAAFzUkdCAK7OHOkAAADUExURefk5w+ruswAAAAFSURBVfJD7cExAQAAAMKg9U9tCU+gAAAAAAAAAIC3AR
+QAAFPIUGoAAAAEIFTkSuQmCC',
  obj_ok: '//img.alicdn.com/tfs/TB1rmyTltfJ8KJy0FeXXXKEXXa-50-74.png',
  bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCSVXXXXXasXFXXXXXXXXXXXX-100-80.png',
  obj_error: '//img.alicdn.com/tfs/TB1q9yTltfJ8KJy0FeXXXKEXXa-50-74.png',
  bg_back_fail: '//img.alicdn.com/tfs/TB1w2oOSFXXXXb4XpXXXXXXXXXXXX-100-80.png',
  upLang:{'cn':{
    _gk_guide: "请摁住鼠标左键，刮出两面盾牌",
    _gk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",
    _gk_loading: "加载中",
    _gk_fail: ['呀，盾牌不见了<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
    _gk_action_timeout: ['我等得太久啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
    _gk_net_err: ['网络实在不给力<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
    _gk_too_fast: ['您刮得太快啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题']
  }}

```

```

}
},
callback: function(data) { //成功回调
window.console && console.log(nc_token)
window.console && console.log(data.sessionId)
window.console && console.log(data.sig)
},
failCallback: function(data) { //拦截or失败回调
},
error: function(data) { //异常回调
}
};
var nc = new noCaptcha(NC_Opt);

</script>
</body>
</html>

```

## 参数详细说明

参数	含义
renderTo	声明刮刮卡需要渲染的目标元素ID
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
type	请写死" scrape"
token	刮刮卡交互验证码的主键，请务必不需要写死固定值。请确保每个用户每次打开页面时，token都是不同的。建议格式为" 您的appkey" + " 时间戳" + " 随机数"
width	刮刮卡长度，最小值300，默认值300
height	刮刮卡宽度，最小值100，默认值100
trans	业务键字段。您可以不写此字段，也可以按照下文中" 问题排查与错误码" 部分文档配置此字段，便于线上问题排查
elementID	通过Dom的ID属性自动填写trans业务键，您可以不写此字段，也可以按照下文中" 问题排查与错误码" 部分文档配置此字段，便于线上问题排查
is_Opt	是否自己配置底层采集组件。如无特殊场景请保持写0或不写此项。默认为0
isEnabled	是否启用，无特殊场景请默认写true。默认为true
language	语言。详细配置方法请见下方" 自定义文案" 部分。默认为cn(中文)

times	允许服务器超时重复次数，默认5次
objects	特殊字段，请务必写死 [ “//img.alicdn.com/tps/TB1BT9jPFXXXXbyXFXXXXXXXXXX-80-80.png” ]
apimap	用于自己指定滑动验证各项请求的接口地址。如无特殊情况，请不要配置这里的apimap
elements	两个长度的List，自定义需要刮出的两个elements的图片资源
bg_back_prepared	自定义刮动时的背景图像，平铺填充
bg_front	自定义刮动的前景（浮层），注意这里只接受base64数据流
obj_ok	自定义验证通过时显示的图标资源
bg_back_pass	自定义验证通过时的背景图像，平铺填充
obj_error	自定义验证失败或异常时显示的图标资源
bg_back_fail	自定义验证失败或异常时显示的背景图像，平铺填充
upLang	用于自定义文案，具体参见下方“自定义文案”部分文档
callback	验证成功时会触发此回调
failCallback	拦截或验证失败时会触发此回调
error	异常回调

## 方法详细说明

无痕验证维护了三个全局方法供您使用，当nvc.js资源加载完成后，这些方法您就可以直接调用了。

方法名	参数	作用
getNVCVal	无	在发送业务请求的前一刻调用此方法，并将此方法返回的信息串随业务请求带去服务端，由服务端使用此信息串调用风控服务查询人机风险
getNC	无	调用此方法会在您NVC_Opt配置项中
show	无	在调用hide方法后，任何时候调用此方法，可以展示刮刮卡组件

# 测试指南

## (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
{ "key1" : " code0" }	刮动一定会通过
{ "key1" : " code300" }	刮动一定会被拦截

## (2) 正式上线测试

正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：·请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写

·请注意您当前验证码的使用场景。如果您的验证码在PCweb场景下使用，请不要用手机等移动设备进行测试

·在测试低版本IE浏览器时，不要用IE11浏览器的兼容模式去模拟生成低版本浏览器。请用真正低版本IE浏览器进行测试

# 自定义文案

您可以通过配置刮刮卡配置项中的upLang参数来自定义您需要的文案。

```
upLang:{'cn':{
  _ggk_guide: "请摁住鼠标左键，刮出两面盾牌",//初始化完成
  _ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",//验证成功
  _ggk_loading: "加载中",//加载过程
  _ggk_fail: ['呀，盾牌不见了<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//验证失败
  _ggk_action_timeout: ['我等得太久啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//操作超时
  _ggk_net_err: ['网络实在不给力<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//加载失败
  _ggk_too_fast: ['您刮得太快啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],//刮动过快
  }
}
```

对于默认的“盾牌”样式的交互与UI，我们为您准备好了18国翻译，您可以根据自己的需求进行替换。替换方法：将language参数和upLang的第一个参数替换为您想要的语言，并在第二个参数中自定义默认UI(盾牌)的

文案(“加载中”、“网络异常”、“刮动过快”、“等的太久”在多语言场景不需要自定义文案，修改 language 参数后会自动提供翻译)

ar\_SA:阿拉伯文

de\_DE:德文

es\_ES:西班牙文

fr\_FR:法文

in\_ID:印尼语

it\_IT:意大利文

iw\_HE:希伯来语

ja\_JP:日文

ko\_KR:韓文

nl\_NL:荷蘭文

pt\_BR:波蘭文

ru\_RU:俄文

th\_TH:泰文

tr\_TR:土耳其文

vi\_VN:越南文

cn:简体中文

tw:繁体中文

en:英文

中文	请摁住鼠标左键，刮出两面盾牌	呀，盾牌不见了，请再来一次或反馈问题	恭喜您成功刮出盾牌，继续下一步操作吧
台湾繁体	請按住滑鼠左鍵，刮出兩個盾牌。	哎呀，盾牌消失了！請再來一次或通報錯誤。	恭喜！您刮出盾牌。請繼續下一步吧！
English	Left click and hold to scrape out two shields.	Uh oh, the shield disappeared! Please try again or report an error.	Congratulations! You gained a shield. Continue to the next step!
Spanish	Mantén pulsado el botón izquierdo del ratón para raspar dos escudos.	¡Vaya! El escudo ha desaparecido. Vuelve a intentarlo o informa del problema.	¡Enhorabuena! Has conseguido un escudo. Continúa con el siguiente paso.
Polish	Kliknij i przytrzymaj lewy przycisk myszy, aby zdrapać dwie tarcze.	Ojej, tarcza znikła! Spróbuj ponownie lub zgłoś problem.	Gratulacje! Zdobywasz tarczę. Przejdź do kolejnego kroku!
French	Maintenez le clic gauche enfoncé pour gratter deux écus.	Oups, l'écu a disparu ! Veuillez réessayer ou signaler une erreur.	Félicitations ! Vous avez reçu un écu. Passez à l'étape suivante !
German	Klicken Sie mit der linken Maustaste und halten Sie sie gedrückt, um zwei Schilde zu kratzen.	Hoppla, der Schild ist verschwunden! Versuchen Sie es erneut oder melden Sie einen Fehler.	Glückwunsch! Sie haben einen Schild gewonnen. Gehen Sie weiter zum nächsten Schritt!

Italian	Fare clic con il tasto sinistro del mouse e tenere premuto per grattare due scudi.	Oh oh, lo scudo è scomparso! Riprovare o segnalare un errore.	Congratulazioni! Hai guadagnato uno scudo. Continua con il passaggio successivo.
Russian	Нажмите и удерживайте левую кнопку мыши, чтобы стереть два щита.	Ох, щит пропал! Повторите попытку или сообщите об ошибке.	Поздравляем! Вы получили щит. Перейдите на следующий шаг!
Japanese	クリックして押し続け、2つのシールドを掘り出してください。	シールドが消えました！再度お試しになるか、エラーを報告してください。	おめでとうございます！シールドを獲得しました。次のステップに進んでください！
Korean	.	! .	! . !
Arabic	.	! .	! . !
Turkish	İki kalkanı kazımak için sol tıklayın ve basılı tutun.	Eyvah, kalkan kayboldu! Lütfen daha sonra tekrar deneyin veya bir hata bildiriminde bulunun.	Tebrikler! Bir kalkan kazandınız. Bir sonraki adıma geçin!
Thai		!	! !
Vietnamese	Nhấp giữ chuột trái để kéo ra hai khiên.	, khiến đã bin mt! Vui lòng th li hoc báo cáo li.	Xin chúc mng! Bn đã giành đc mt khiên. Tip tc bc tip theo!
Dutch	Houd de linkermuisknop ingedrukt om twee schilden weg te strepen.	Oh, het schild is verdwenen! Probeer het opnieuw of rapporteer uw probleem.	Gefeliciteerd! U hebt een schild gekregen. Ga verder met de volgende stap!
Hebrew	.	-, ! .	! . !
Indonesian	Klik kiri dan tahan untuk menggrurat dua buah perisai.	Oh, perisainya menghilang! Harap coba lagi atau laporkan kesalahan.	Selamat! Anda mendapatkan sebuah perisai. Lanjutkan ke langkah berikutnya!

## 问题排查

### (1) 上传业务键

刮刮卡交互验证默认不会采集您当前业务场景下的用户名、手机号等业务键，为了方便问题排查，这里强烈建议您配置上传业务主键，这样遇到问题的客户可以提供“业务键+时间”，很方便的关联到对应刮刮卡的请求

。否则您必须提供引导用户进行抓包才能具体对应查到日志和请求。您可以配置elementID来让滑动验证主动采集。elementID的值为一个stringList，配置之后在刮动结束的时刻，滑块会以elementID当中的每一个string作为ID，去获取对应dom的value值并上传。举个例子，如果您配置elementID:[ "usernameID" ]，则刮动结束时document.getElementById( "usernameID" ).value会自动被上传至验证码服务端。

## (2) 获取token

如果您不方便上传业务键，在用户遇到问题时您可以引导其点击“反馈问题”链接，然后将打开页面的url复制保存下来以方便后续查询日志，无需您自己进行抓包。

# 滑动验证

## H5版本

## 简介

滑动验证是服役于阿里巴巴人机识别第一线的验证码产品。用户通过简单的右滑交互，无需思考即可通过验证





登录

淘宝及1688会员可直接使用会员名登录

登录名：

fmwtest

登录密码：

[忘记密码](#)

登录密码

滑动验证码：

向右滑动验证

登录

[免费注册](#)

其他方式登录



# 资源引入说明

使用H5版滑动验证码，需要引入以下js资源：

```
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/nch5/index.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8"
src="//aeis.alicdn.com/sd/nch5/index.js?t=2015052012"></script> -->
```

其中，t字段建议设置为小时级别时间戳，随时间更新，确保js不会被浏览器长时间缓存，这样能够享受到我们针对js的动态发布迭代。

## 完整接入demo

```
<html>
<head>
<meta charset="utf-8"/>
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0,
user-scalable=0">
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/nch5/index.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8"
src="//aeis.alicdn.com/sd/nch5/index.js?t=2015052012"></script> -->
</head>
<body>
<div id="__nc" style="margin-left:auto;margin-right:auto;width:80%;height:100px;padding-top:100px;">
<div id="nc"></div>
</div>

<script>
var nc_token = ["CF_APP_1", (new Date()).getTime(), Math.random()].join(':');
var nc=NoCaptcha.init({
  renderTo: '#nc',
  appkey: 'CF_APP_1',
  scene: 'register',
  token: nc_token,
  trans: {"key1": "code200"},
  elementID: ["usernameID"],
  is_Opt: 0,
  language: "cn",
  timeout: 10000,
  retryTimes: 5,
  errorTimes: 5,
  inline:false,
  apimap: {
    // 'analyze': '//a.com/nocaptcha/analyze.jsonp',
    // 'uab_Url': '//aeu.alicdn.com/js/uac/909.js',
```

```

},
bannerHidden:false,
initHidden:false,
callback: function (data) {
window.console && console.log(nc_token)
window.console && console.log(data.csessionid)
window.console && console.log(data.sig)
},
error: function (s) {
}
});
NoCaptcha.setEnabled(true);
nc.reset();//请务必确保这里调用一次reset()方法

NoCaptcha.upLang('cn', {
'LOADING':"加载中..."//加载
'SLIDER_LABEL': "请向右滑动验证"//等待滑动
'CHECK_Y':"验证通过"//通过
'ERROR_TITLE':"非常抱歉，这出错了..."//拦截
'CHECK_N':"验证未通过", //准备唤醒二次验证
'OVERLAY_INFORM':"经检测你当前操作环境存在风险，请输入验证码"//二次验证
'TIPS_TITLE':"验证码错误，请重新输入"//验证码输错时的提示
});
</script>
</body>
</html>

```

## 参数详细说明

参数	含义	是否必填项
renderTo	声明滑动验证需要渲染的目标元素ID	是
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写	是
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写	是
token	滑动验证码的主键，请务必不需要写死固定值。请确保每个用户每次打开页面时，token都是不同的。建议格式为“您的appkey” + “时间戳” + “随机数”	是
trans	业务键字段。您可以不写此字段，也可以按照下文中“问题排查与错误码”部分文档配置此字段，便于线上问题排查	否

elementID	通过Dom的ID属性自动填写trans业务键，您可以不写此字段，也可以按照下文中“问题排查与错误码”部分文档配置此字段，便于线上问题排查	否
is_Opt	是否自己配置底层采集组件。如无特殊场景请保持写0或不写此项。默认为0	否
language	语言。H5场景默认支持简中、繁中、英文，详细配置方法请见下方“自定义文案与多语言”部分。默认为cn(中文)	否
timeout	内部网络请求的超时时间，一般不需要改，默认值为10000ms	否
retryTimes	允许服务器超时重复次数，默认5次	否
errorTimes	弹出二次验证时，允许二次验证填错的次数，默认5次	否
bannerHidden	验证通过后，验证码组件是否自动隐藏，默认为 true	否
initHidden	是否默认不渲染，设置为true时，不会自动渲染，需要自行调用show方法将其渲染，默认为 false	否
inline	弹出二次验证是否为inline方式，默认为false(浮层模式)	否
apimap	用于自己指定滑动验证各项请求的接口地址。如无特殊情况，请不要配置这里的apimap	否
callback	前端滑动验证通过时会触发此回调，您可以在这个回调中将token、sessionid、sig记录下来，随业务请求带到您的server端调用验签	是

## 方法详细说明

NoCaptcha的实例化对象中维护了一些方法提供给您使用，您可以通过诸如var nc=NoCaptcha.init(NC\_Opt); nc.functionName(parm);的方式来调用使用它们。

方法名	参数	作用
reset	无	任何时候调用此方法，可以将滑动验证重置为初始状态(等待滑动)
hide	无	任何时候调用此方法，可以隐藏

		滑动验证组件
show	无	在调用hide方法后，任何时候调用此方法，可以展示滑动验证组件
setTrans	Object	可以设置滑动验证的业务键，方便问题排查定位。详见“问题排查与错误码说明”部分
upLang	String,Object	用于配置NC自定义文案，详见“自定义文案与多语言”部分
on	事件名(String),回调函数(Function)	为滑动验证状态设置监听

为滑动验证设置事件监听的具体方法：  
var nc=NoCaptcha.init(NC\_Opt); nc.on( '事件名' , 回调函数);

事件列表：

事件名	含义
beforeverify	弹出验证浮层之前
afterverify	弹出验证浮层之后
error	系统或者网络错误
fail	用户验证失败
success	用户验证成功
refresh	点击刷新

## 测试指南

### (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or二次验证or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

CF\_APP\_1下trans字段与验证码结果对应关系如下：

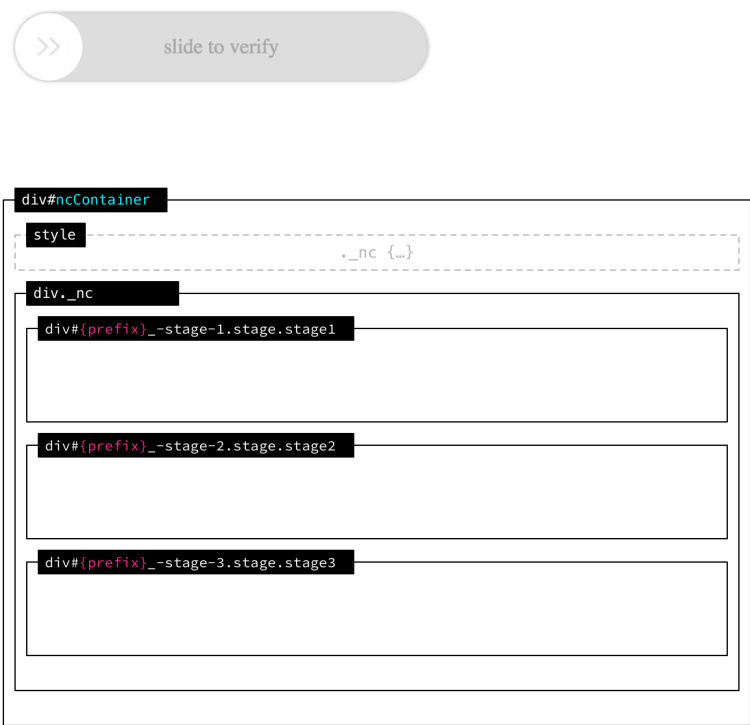
trans	效果
{ "key1" : " code0" }	滑动一定会通过
{ "key1" : " code200" }	滑动一定出现二次验证
{ "key1" : " code300" }	滑动一定会被拦截

## (2) 正式上线测试

- 正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：
- 请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写
  - 请注意您当前验证码的使用场景。如果您的验证码在HTML5场景下使用，请不要用PCWeb浏览器进行测试

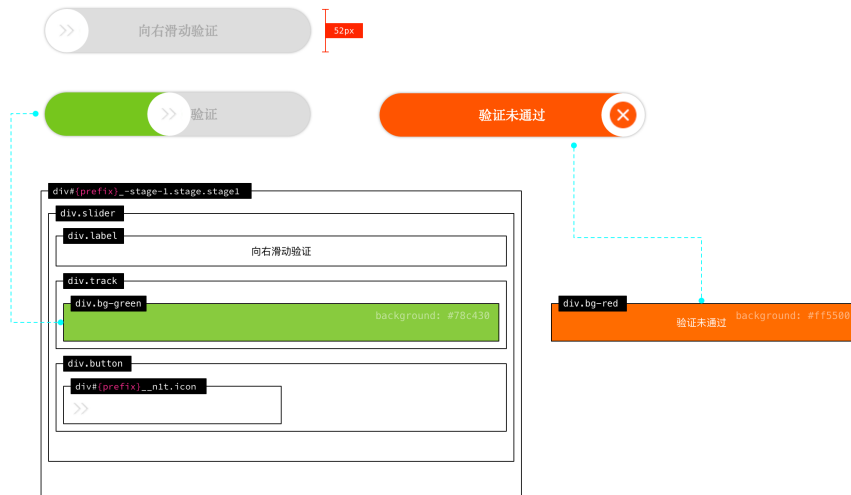
## 自定义样式：HTML结构与CSS定制说明

滑动验证Html5环境下的Dom结构如下图所示,共分为3种stage状态：stage1(初始状态)、stage2(二次验证)、stage3(拦截状态)。其中 {prefix} 为前缀，值形如 nc\_1、nc\_2...



您可以通过自定义CSS的方式来覆盖滑动验证各个UI组件的颜色、资源等，注意自定义样式需出现在滑动验证自带样式的后方，必要时可添加 `!important`

stage1(初始状态)：



```

/* 滑动条高度、边框、背景色等 */
.nc .stage1 .slider {
  height: 52px;
  border-radius: 26px;
  box-shadow: 0 0 3px #999;
  background-color: #ddd;
}
/* 滑动条 */
.nc .stage1 .track div {
  border-radius: 26px;
  color: #fff;
}
/* 滑动条背景色-正常 */
.nc .stage1 .bg-green {
  background-color: #78c438;
}
/* 滑动条背景色-失败 */
.nc .stage1 .bg-red {
  background-color: #ff5500;
}
/* 各个状态的icon */
等待滑动: "icon iconfont icon-slide-arrow"
验证出错: "icon no iconfont icon-close"
验证通过: "icon yes iconfont icon-ok"

```

stage2(二次验证) :

验证码

7kXF

确定

abcd

确定

经检测你当前操作环境存在风险，请输入验证码

验证码

2K8

确定

验证码

nArd

确定

验证码错误，请重新输入

div.nc.nc-h5-overlay

div.info

i.iconfont.icon-warn

span

经检测你当前操作环境存在风险，请输入验证码

div#(prefix)\_.stage-2.stage.stage2

div.textbox

input[type=text]

div.placeholder

验证码

div.icon.tip

div.icon.clear

div.icon.refresh

img.code

FTiY

button.btn-ok

确定

div.tips

验证码错误，请重新输入

```
.nc-h5-overlay div.info i.iconfont {
color: #f00;
width: 1.2em;
font-size: 120%;
margin-right: .3em;
```



```
}
/* 验证码输入框 */
._nc .stage2 .textbox .input {
background: none;
width: 140px;
height: 40px;
outline: none;
font-size: 18px;
color: #333;
}
/* 图标 */
._nc .icon {
width: 32px;
height: 32px;
background-image: url("//g.alicdn.com/sd/nch5/icons.png?v=2");
}
._nc .stage2 .icon.tip {
left: 10px;
background-position: -192px 0;
}
._nc .stage2 .icon.clear {
left: 106px;
background-position: -224px 0;
}
._nc .stage2 .icon.refresh {
right: 11px;
background-position: -256px 0;
}
/* 确定按钮、提示 */
._nc .stage2 .btn-ok, ._nc .stage2 .tips {
width: 100%;
height: 46px;
border-radius: 5px;
color: #fff;
font-size: 20px;
border: none;
outline: none;
background-color: #3199f4;
}
._nc .stage2 .btn-ok:disabled {
background-color: #ccc;
}
._nc .stage2 .tips {
background-color: #333;
color: #fff;
text-align: center;
line-height: 46px;
}
```

stage3(拦截状态) :

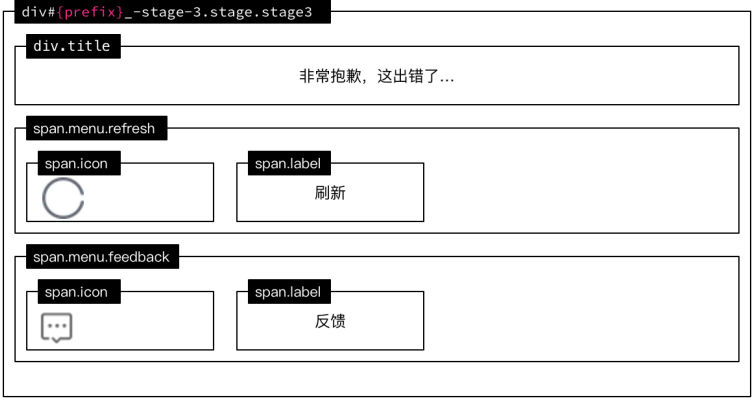
非常抱歉，这出错了...



刷新



反馈



```
._nc .stage3 .title {
font-size: 20px;
color: #777;
width: 100%;
}
/* 图标 */
._nc .icon {
width: 32px;
height: 32px;
background-image: url("//g.alicdn.com/sd/nch5/icons.png?v=2");
}
._nc .stage3 .menu.refresh .icon {
background-position: -128px 0;
}
._nc .stage3 .menu.feedback .icon {
background-position: -160px 0;
}
/* 刷新链接 */
._nc .stage3 .menu.refresh {
border-right: 2px solid #ccc;
}
```

# 自定义文案与多语言

## (1) 多语言

滑动验证默认支持简体中文、繁体中文、英文三种语言，您可以通过滑动验证配置项中的language字段来进行选择（默认为cn，简体中文）

cn:简体中文

tw:繁体中文

en:英文

## (2) 自定义文案

您可以通过滑动验证对象维护的upLang方法来自定义您需要的文案。内置的语言有简体中文、繁体中文、英文（cn、tw、en）三种，可以选择其中一种进行更新，也可传入一个新名字以建立新语言。如果您调用upLang方法后，自定义文案没有生效，请在其之后调用reset()方法，让滑动验证重新渲染即可

```
/**
 * @param name {string} 语言名，可选 cn、tw、en，也可添加自定义语言
 * @param detail {object} 语言详情，key/value 格式
 */
NoCaptcha.upLang('cn', {
  'LOADING': "加载中..." // 加载
  'SLIDER_LABEL': "请向右滑动验证" // 等待滑动
  'CHECK_Y': "验证通过" // 通过
  'ERROR_TITLE': "非常抱歉，这出错了..." // 拦截
  'CHECK_N': "验证未通过" // 准备唤醒二次验证
  'OVERLAY_INFORM': "经检测您当前操作环境存在风险，请输入验证码" // 二次验证
  'TIPS_TITLE': "验证码错误，请重新输入" // 验证码输错时的提示
});
```

# 问题排查与错误码说明

## (1) 上传业务键

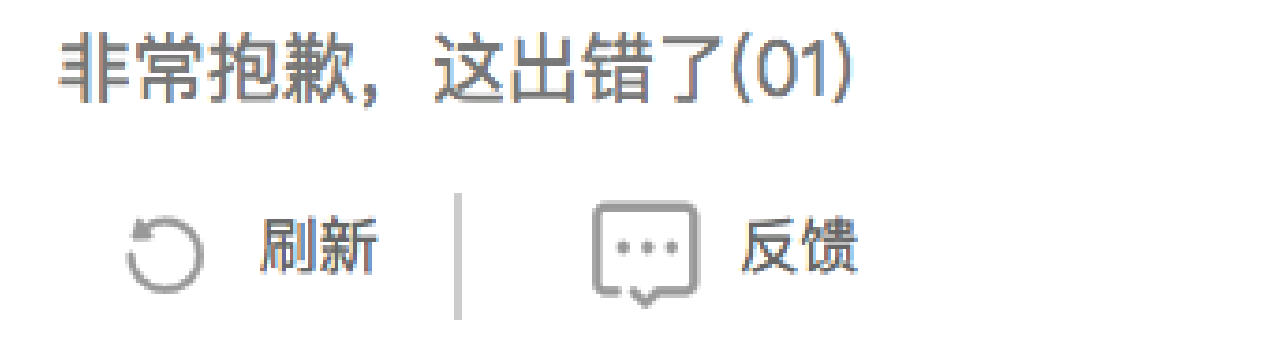
滑动验证默认不会采集您当前业务场景下的用户名、手机号等业务键，为了方便问题排查，这里强烈建议您配置上传业务主键，这样遇到问题的客户可以提供“业务键+时间”，很方便的关联到对应滑动验证的请求。否则您必须提供引导用户进行抓包才能具体对应查到日志和请求。

您有两种方法可以上传业务键：

- 您可以在用户操作滑块前的任何时刻调用滑动验证维护的setTrans方法上传。比如您可以在用户填写完用户名（从用户名输入框失焦）时调用nc.setTrans({ 'userName': 用户名输入框value})来上传用户名；也可以在页面加载完毕的时候调用nc.setTrans({ '业务会话标示': 业务会话session的值})来上传您业务上唯一的会话标示。
- 您也可以配置elementID来让滑动验证主动采集。elementID的值为一个stringList，配置之后在滑动结束的时刻，滑块会以elementID当中的每一个string作为ID，去获取对应dom的value值并上传。举个例子，如果您配置elementID: [ "usernameID" ]，则滑动结束时document.getElementById( "usernameID" ).value会自动被上传至滑动验证服务端。

## (2) 错误码

滑动验证发生异常的情况下，在错误信息后会附带错误码，如下图：



其中错误码含义如下：

错误码	含义
00	analyze请求超时
01	uab.js加载超时
B0	策略拦截

## PC-WEB版本

### 简介

滑动验证是服役于阿里巴巴人机识别第一线的验证码产品。用户通过简单的右滑交互，无需思考即可通过验证



# 资源引入说明

使用PC版滑动验证码，需要引入以下js资源：

```
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8" src="//aeis.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
-->
```

其中，t字段建议设置为小时级别时间戳，随时间更新，确保js不会被浏览器长时间缓存，这样能够享受到我们针对js的动态发布迭代。

## 完整接入demo

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<!-- 国内使用 -->
<script type="text/javascript" charset="utf-8" src="//g.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
<!-- 若您的主要用户来源于海外，请替换使用下面的js资源 -->
<!-- <script type="text/javascript" charset="utf-8" src="//aeis.alicdn.com/sd/ncpc/nc.js?t=2015052012"></script>
-->
</head>
<body>
<div id="your-dom-id" class="nc-container"></div> <!--No-Captcha渲染的位置，其中 class 中必须包含 nc-
container-->

<script type="text/javascript">
var nc_token = ["CF_APP_1", (new Date()).getTime(), Math.random()].join(':');
var NC_Opt =
{
  renderTo: "#your-dom-id",
  appkey: "CF_APP_1",
  scene: "register",
  token: nc_token,
  customWidth: 300,
  trans:{"key1":"code0"},
  elementID: ["usernameID"],
  is_Opt: 0,
  language: "cn",
  isEnabled: true,
  timeout: 3000,
  times:5,
  apimap: {
    // 'analyze': '//a.com/nocaptcha/analyze.jsonp',
    // 'get_captcha': '//b.com/get_captcha/ver3',
    // 'get_captcha': '//pin3.aliyun.com/get_captcha/ver3'
```

```
// 'get_img': '//c.com/get_img',
// 'checkcode': '//d.com/captcha/checkcode.jsonp',
// 'umid_Url': '//e.com/security/umscript/3.2.1/um.js',
// 'uab_Url': '//aeu.alicdn.com/js/uac/909.js',
// 'umid_serUrl': 'https://g.com/service/um.json'
},
callback: function (data) {
window.console && console.log(nc_token)
window.console && console.log(data.csessionid)
window.console && console.log(data.sig)
}
}
var nc = new noCaptcha(NC_Opt)
nc.upLang('cn', {
_startTEXT: "请按住滑块，拖动到最右边",
_yesTEXT: "验证通过",
_error300: "哎呀，出错了，点击<a href=\"javascript:__nc.reset()\">刷新</a>再来一次",
_errorNetwork: "网络不给力，请<a href=\"javascript:__nc.reset()\">点击刷新</a>",
})
</script>
</body>
</html>
```

## 参数详细说明

参数	含义	是否必填项
renderTo	声明滑动验证需要渲染的目标元素ID	是
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写	是
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写	是
token	滑动验证码的主键，请务必不需要写死固定值。请确保每个用户每次打开页面时，token都是不同的。建议格式为“您的appkey” + “时间戳” + “随机数”	是
customWidth	滑动条的长度，建议预留300像素以上	否
trans	业务键字段。您可以不写此字段，也可以按照下文中“问题排查与错误码”部分文档配置此字段，便于线上问题排查	否

elementID	通过Dom的ID属性自动填写trans业务键，您可以不写此字段，也可以按照下文中“问题排查与错误码”部分文档配置此字段，便于线上问题排查	否
is_Opt	是否自己配置底层采集组件。如无特殊场景请保持写0或不写此项。默认为0	否
language	语言。PC场景默认支持18国语言，详细配置方法请见下方“自定义文案与多语言”部分。默认为cn(中文)	否
isEnabled	是否启用，无特殊场景请默认写true。默认为true	否
timeout	内部网络请求的超时时间，一般不需要改，默认值为3000ms	否
times	允许服务器超时重复次数，默认5次	否
apimap	用于自己指定滑动验证各项请求的接口地址。如无特殊情况，请不要配置这里的apimap	否
callback	前端滑动验证通过时会触发此回调，您可以在这个回调中将token、sessionid、sig记录下来，随业务请求带到您的server端调用验签	是

## 方法详细说明

noCaptcha对象中维护了一些方法提供给您使用，您可以通过诸如var nc = new noCaptcha(NC\_Opt); nc.functionName(parm);的方式来调用使用它们。

方法名	参数	作用
reload	无	任何时候调用此方法，可以将滑动验证重置为初始状态(等待滑动)
hide	无	任何时候调用此方法，可以隐藏滑动验证组件
show	无	在调用hide方法后，任何时候调用此方法，可以展示滑动验证组件
getToken	无	任何时候调用此方法，可以获取当前滑动验证实例的token
setTrans	Object	可以设置滑动验证的业务键，方便问题排查定位。详见“问题排

		查与错误码说明” 部分
upLang	String,Object	用于配置NC自定义文案，详见“ 自定义文案与多语言” 部分
destroy	无	销毁当前滑动验证实例

## 测试指南

### (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
{ "key1" : " code0" }	滑动一定会通过
{ "key1" : " code300" }	滑动一定会被拦截

### (2) 正式上线测试

正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：

- 请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写
- 请注意您当前验证码的使用场景。如果您的验证码在PCweb场景下使用，请不要用手机等移动设备进行测试
- 在测试低版本IE浏览器时，不要用IE11浏览器的兼容模式去模拟生成低版本浏览器。请用真正低版本IE浏览器进行测试

## 自定义样式：HTML结构与CSS定制说明

### (1) HTML结构图示

滑动验证PCweb环境下的Dom结构如下图所示



快速登录

账户密码登录

手机号/会员名/邮箱

>> 请按住滑块，拖动到最右边

250px

34px

忘记密码?

免费注册

登 录

div#nocaptch.tb-login.nc-container

div#(prefix)\_nlt.nc\_scale

div#(prefix)\_bg.nc\_bg

span#(prefix)\_niz

div#(prefix)\_scal\_text.scal\_text

span.nc-lang-cnt

div#(prefix)\_clickCaptcha

div#(prefix)\_imgCaptcha

img#(prefix)\_voicebtn

b#(prefix)\_helpbtn

div#(prefix)\_nc\_voice

说明：此处的元素 id #nocaptcha 不是固定值，由实际引用者指定。前缀 {prefix} 的值形如：nc\_1、nc\_2 ...

span#\_n12.btnok

40



## (2) 颜色、图标自定义

您可以通过自定义CSS的方式来覆盖滑动验证各个UI组件的颜色、资源等，注意自定义样式需出现在滑动验证自带样式的后方，必要时可添加 !important



```
.nc_scale {
background: #e8e8e8; /* 默认背景色 */
}
.nc_scale .scale_text {
color: #9c9c9c; /* 默认字体颜色 */
}
.nc_scale div.nc_bg {
background: #7ac23c; /* 滑过时的背景色 */
}
```

```
}
.nc_scale .scale_text2 {
color: #fff; /* 滑过时的字体颜色 */
}

.nc_scale span {
border: 1px solid #ccc;
background: #fff url("//g.alicdn.com/sd/ncpc/images/rt.png") no-repeat center;
}
.nc_scale .btnok {
background: #fff url("//g.alicdn.com/sd/ncpc/images/yes.png") no-repeat center;
}
.errloading {
border: #faf1d5 1px solid;
background: url("//g.alicdn.com/sd/ncpc/images/no.png") #ffffff no-repeat 10px 9px;
color: #ef9f06;
}
```

## 自定义文案与多语言

### (1) 多语言

滑动验证默认支持18国语言，您可以通过滑动验证配置项中的language字段来进行选择（默认为cn，中文）

ar\_SA:阿拉伯文  
de\_DE:德文  
es\_ES:西班牙文  
fr\_FR:法文  
in\_ID:印尼语  
it\_IT:意大利文  
iw\_HE:希伯来语  
ja\_JP:日文  
ko\_KR:韩文  
nl\_NL:荷兰文  
pt\_BR:波兰文  
ru\_RU:俄文  
th\_TH:泰文  
tr\_TR:土耳其文  
vi\_VN:越南文  
cn:简体中文  
tw:繁体中文  
en:英文

### (2) 自定义文案

您可以通过滑动验证对象维护的upLang方法来自定义您需要的文案。内置的语言有简体中文、繁体中文、英文

( cn、tw、en ) 三种，可以选择其中一种进行更新，也可传入一个新名字以建立新语言。如果您调用 upLang方法后，自定义文案没有生效，请在其之后调用reload()方法，让滑动验证重新渲染即可

```
/**
 * @param name {string} 语言名，可选 cn、tw、en ，也可添加自定义语言
 * @param detail {object} 语言详情，key/value 格式
 */
nc.upLang('cn',{
  _startTEXT: "请按住滑块，拖动到最右边",
  _yesTEXT: "验证通过",
  _error300: "哎呀，出错了，点击<a href=\"javascript:__nc.reset()\">刷新</a>再来一次",
  _errorNetwork: "网络不给力，请<a href=\"javascript:__nc.reset()\">点击刷新</a>",
})
```

## 问题排查与错误码说明

### (1) 上传业务键

滑动验证默认不会采集您当前业务场景下的用户名、手机号等业务键，为了方便问题排查，这里强烈建议您配置上传业务主键，这样遇到问题的客户可以提供“业务键+时间”，很方便的关联到对应滑动验证的请求。否则您必须提供引导用户进行抓包才能具体对应查到日志和请求。

您有两种方法可以上传业务键：

- a. 您可以在用户操作滑块前的任何时刻调用滑动验证维护的setTrans方法上传。比如您可以在用户填写完用户名（从用户名输入框失焦）时调用nc.setTrans({ 'userName':用户名输入框value})来上传用户名；也可以在页面加载完毕的时候调用nc.setTrans({ '业务会话标示':业务会话session的值})来上传您业务上唯一的会话标示。
- b. 您也可以配置elementID来让滑动验证主动采集。elementID的值为一个stringList，配置之后在滑动结束的時刻，滑块会以elementID当中的每一个string作为ID，去获取对应dom的value值并上传。举个例子，如果您配置elementID:[ "usernameID" ]，则滑动结束时document.getElementById( "usernameID" ).value会自动被上传至滑动验证服务端。

### (2) 错误码

滑动验证发生异常的情况下，在错误信息后会附带错误码，如下图：



其中错误码含义如下：

错误码	含义
00	analyze请求超时
01	uab.js加载超时
02	um.js加载超时

04	设备指纹代码加载异常
08	waitForUmx超出重试次数

# 无痕验证

## H5版本

### 简介

无痕验证是阿里巴巴提供的新一套人机解决方案。由无痕组件负责收集和统一调度下游验证码产品实现综合人机对抗。

用户接入后完全静默采集，对大部分正常用户0打扰，对有风险用户会根据风险等级返回相应的下游验证码产品唤醒建议。

无痕验证采用服务端-服务端的风险查询机制，业务方在有风控能力的情况下可以很轻松的将无痕验证人机识别的能力并联到自己的业务风控上。

### 资源引入说明

使用无痕验证，需要引入以下js资源：

```
<script type="text/javascript" charset="utf-8"
src="//g.alicdn.com/sd/nvc/1.1.112/guide.js?t=2015052012"> </script>
```

其中，t字段建议设置为小时级别时间戳，随时间更新，确保js不会被浏览器长时间缓存，这样能够享受到我们针对js的动态发布迭代。

### 完整接入demo

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0,
user-scalable=0">
</head>
<body>
<div id="captcha"></div>
<button type="button" id='register'>注册</button>

<script>
window.NVC_Opt = {
//无痕配置 && 滑动验证、刮刮卡通用配置
appkey:'CF_APP_1',
scene:'nvc_register',
isH5:true,
popUp:false,
renderTo:'#captcha',
trans: {"key1": "code0", "nvcCode":400},
language: "cn",
//滑动验证长度配置
customWidth: 300,
//刮刮卡配置项
width: 300,
height: 100,
elements: [
'//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png',
'//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png'
],
bg_back_prepared: '//img.alicdn.com/tps/TB1skE5SFXXXb3XXXXXXXXXXXX-100-80.png',
bg_front:
'data:image/png;base64,iVBORw0KGGoAAAANSUheUgAAAGQAAABQCAMAAADY1yDdAAAAABGdBTUEAALGPC/xhB
QAAAFzUkdCAK7OHOkAAADUExURefk5w+ruswAAAAFSURBVfjD7cExAQAAAMKg9U9tCU+gAAAAAAAAAIC3AR
+QAAFPIUGoAAAAAEIFTkSuQmCC',
obj_ok: '//img.alicdn.com/tfs/TB1rmyTltfJ8KJy0FeXXXKEXXa-50-74.png',
bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCSVXXXasXFXfXXXXXXXXXXXX-100-80.png',
obj_error: '//img.alicdn.com/tfs/TB1q9yTltfJ8KJy0FeXXXKEXXa-50-74.png',
bg_back_fail: '//img.alicdn.com/tfs/TB1w2oOSFXXXb4XpXXXXXXXXXXXX-100-80.png',
upLang:{"cn":{"
_ggk_guide: "请在屏幕上滑动，刮出两面盾牌",
_ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",
_ggk_loading: "加载中",
_ggk_fail: ['呀，盾牌不见了<br/>请', "javascript:NoCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_action_timeout: ['我等得太久啦<br/>请', "javascript:NoCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_net_err: ['网络实在不给力<br/>请', "javascript:NoCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_too_fast: ['您刮得太快啦<br/>请', "javascript:NoCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题']
}
}
}

function yourRegisterRequest(url, params){
var callbackName = ('jsonp_' + Math.random()).replace('.', '')
params += '&callback=' + callbackName
var o_scripts = document.getElementsByTagName("script")[0]
```

```

var o_s = document.createElement('script')
o_scripts.parentNode.insertBefore(o_s, o_scripts);
//您注册请求的业务回调
window[callbackName] = function(json) {
if(json.result.code == 400) {
//唤醒滑动验证
getNC().then(function(){
NoCaptcha.upLang('cn', {
'LOADING': "加载中..." //加载
'SLIDER_LABEL': "请向右滑动验证" //等待滑动
'CHECK_Y': "验证通过" //通过
'ERROR_TITLE': "非常抱歉，这出错了..." //拦截
'CHECK_N': "验证未通过", //准备唤醒二次验证
'OVERLAY_INFORM': "经检测您当前操作环境存在风险，请输入验证码" //二次验证
'TIPS_TITLE': "验证码错误，请重新输入" //验证码输错时的提示
});
_nvc_nc.reset();
})
} else if (json.result.code == 600) {
//唤醒刮刮卡
getSC().then(function(){} )
} else if (json.result.code == 100 || json.result.code == 200) {
//注册成功
alert("register success!")
} else if (json.result.code == 800 || json.result.code == 900) {
//直接拦截
alert("register failed!")
}
}
o_s.src = url + '?' + params
}

document.getElementById('register').onclick = function() {
var params = 'a=' + getNVCVal()
yourRegisterRequest('http://cf.aliyun.com/nvc/nvcAnalyze.jsonp', params)
}
</script>
<script src="//g.alicdn.com/sd/nvc/1.1.112/guide.js"></script>

</body>
</html>

```

## 接入注意事项

- 1.window.NVC\_Opt全局变量的声明一定要在“[//g.alicdn.com/sd/nvc/1.1.112/guide.js](http://g.alicdn.com/sd/nvc/1.1.112/guide.js)” 资源引入之前配置完毕
- 2.切记：yourRegisterRequest方法中维护的“<http://cf.aliyun.com/nvc/nvcAnalyze.jsonp>” 接口必须顶替为您自己的业务请求接口（现有接口属于线上Demo接口，不具备任何实际攻防能力），在这个接口调用的过程中需带上getNVCVal()获取的人机信息串，随业务请求一起带到您的业务服务端，并按照“Link:服务端接入文档”中的指引发起“业务服务端->风控服务端”的风险查询请求。
- 3.风险查询请求返回的结果，您可以直接返回给客户端使用，客户端可以根据风险等级选择放过、拦截、或者调用getNC()、getSC()方法在指定div上唤醒二次验证。如果您服务端维护有自己的风控策略（比如黑白名单

)，也可以综合您自己风控策略与人机接口二者的结果，综合在客户端决定风险处置方式。

## 参数详细说明

参数	含义
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
isH5	是否是Html5场景，这里H5接入需填true。默认为false
popUp	决定二次验证唤醒的方式。true为浮层式，false为嵌入式
renderTo	声明二次验证需要渲染的目标元素ID
trans	业务键字段。您可以不写此字段，也可以按照下文中“问题排查”部分文档配置此字段，便于线上问题排查
customWidth	当唤醒滑动验证作为二次验证时，配置滑动验证的长度
width	当唤醒刮刮卡验证作为二次验证时，配置刮刮卡的长度
height	当唤醒刮刮卡验证作为二次验证时，配置刮刮卡的高度
language	配置二次验证的语言，详见“自定义文案”部分。默认为cn(中文)
elements	两个长度的List，自定义需要刮出的两个elements的图片资源
bg_back_prepared	自定义刮动时的背景图像，平铺填充
bg_front	自定义刮动的前景（浮层），注意这里只接受base64数据流
obj_ok	自定义验证通过时显示的图标资源
bg_back_pass	自定义验证通过时的背景图像，平铺填充
obj_error	自定义验证失败或异常时显示的图标资源
bg_back_fail	自定义验证失败或异常时显示的背景图像，平铺填充
upLang	用于自定义刮刮卡文案，具体可以参见“刮刮卡交互验证前端代码集成文档”



# 测试指南

## (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or二次验证or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

当您使用demo中提供的<http://cf.aliyun.com/nvc/nvcAnalyze.jsonp> 服务端接口进行测试时，CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
{ "nvcCode" :200}	无痕验证直接通过
{ "nvcCode" :400, "key1" : "code0" }	无痕验证会唤醒滑动验证，且滑动验证一定通过
{ "nvcCode" :400, "key1" : "code300" }	无痕验证会唤醒滑动验证，且滑动验证一定拦截
{ "nvcCode" :600, "key1" : "code0" }	无痕验证会唤醒刮刮卡验证，且刮刮卡验证一定通过
{ "nvcCode" :600, "key1" : "code300" }	无痕验证会唤醒刮刮卡验证，且刮刮卡验证一定拦截
{ "nvcCode" :800}	无痕验证直接拦截

## (2) 正式上线测试

- 正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：
- 请务必将demo中的<http://cf.aliyun.com/nvc/nvcAnalyze.jsonp> 接口替换为您自己的业务接口，并按照无痕验证服务端接入文档进行后端风控接入
  - 请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写
  - 请注意您当前验证码的使用场景。如果您的验证码在Html5场景下使用，请不要用PC浏览器进行测试
  - 在测试低版本IE浏览器时，不要用IE11浏览器的兼容模式去模拟生成低版本浏览器。请用真正低版本IE浏览器进行测试

# 自定义文案

## (1)二次弹出滑动验证文案

调用getNC()方法唤醒滑动验证时，可以在getNC().then(function{})中调用NoCaptcha.upLang方法来自定义文案。自定义之后请记得调用reset()方法刷新渲染滑动验证。

```
getNC().then(function(){
```

```
NoCaptcha.upLang('cn', {
  'LOADING': "加载中..." //加载
  'SLIDER_LABEL': "请向右滑动验证" //等待滑动
  'CHECK_Y': "验证通过" //通过
  'ERROR_TITLE': "非常抱歉，这出错了..." //拦截
  'CHECK_N': "验证未通过", //准备唤醒二次验证
  'OVERLAY_INFORM': "经检测你当前操作环境存在风险，请输入验证码" //二次验证
  'TIPS_TITLE': "验证码错误，请重新输入" //验证码输错时的提示
});
_nvc_nc.reset()
})
```

## (2)二次弹出刮刮卡验证文案

调用getSC()方法唤醒刮刮卡验证时，自定义文案可以直接在NVC\_Opt配置项中配置upLang来修改。

```
upLang:{"cn":{
  _ggk_guide: "请摁住鼠标左键，刮出两面盾牌" //初始化完成
  _ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧" //验证成功
  _ggk_loading: "加载中" //加载过程
  _ggk_fail: ['呀，盾牌不见了<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'] //验证失败
  _ggk_action_timeout: ['我等得太久啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'] //操作超时
  _ggk_net_err: ['网络实在不给力<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'] //加载失败
  _ggk_too_fast: ['您刮得太快啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
    "http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'] //刮动过快
  }
}
```

# PC-WEB版本

## 简介

无痕验证是阿里巴巴提供的新一套人机解决方案。由无痕组件负责收集和统一调度下游验证码产品实现综合人机对抗。

用户接入后完全静默采集，对大部分正常用户0打扰，对有风险用户会根据风险等级返回相应的下游验证码产品唤醒建议。

无痕验证采用服务端-服务端的风险查询机制，业务方在有风控能力的情况下可以很轻松的将无痕验证人机识别的能力并联到自己的业务风控上。

# 资源引入说明

使用无痕验证，需要引入以下js资源：

```
<script type="text/javascript" charset="utf-8"
src="//g.alicdn.com/sd/nvc/1.1.112/guide.js?t=2015052012"> </script>
```

其中，t字段建议设置为小时级别时间戳，随时间更新，确保js不会被浏览器长时间缓存，这样能够享受到我们针对js的动态发布迭代。

## 完整接入demo

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0,
user-scalable=0">
</head>
<body>
<div id="captcha"> </div>
<button type="button" id='register'>注册</button>

<script>
window.NVC_Opt = {
//无痕配置 && 滑动验证、刮刮卡通用配置
appkey:'CF_APP_1',
scene:'nvc_register',
isH5:false,
popUp:false,
renderTo:'#captcha',
trans: {"key1": "code0", "nvcCode":400},
language: "cn",
//滑动验证长度配置
customWidth: 300,
//刮刮卡配置项
width: 300,
height: 100,
elements: [
'//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png',
'//img.alicdn.com/tfs/TB17cwllsLJ8KJy0FnXXcFDpXa-50-74.png'
],
bg_back_prepared: '//img.alicdn.com/tps/TB1skE5SFXXXXb3XXXXXXXXXXXX-100-80.png',
bg_front:
'data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAGQAAABQCAMAAADY1yDdAAAAABGdBTUEAALGPC/xhB
QAAAAFzUkdCAK7OHOkAAADUExURfk5w+ruswAAAAfSURBVfJD7cExAQAAAMKg9U9tCU+gAAAAAAAAAIC3AR
+QAAFPiUGoAAAAAEIFTkSuQmCC',
obj_ok: '//img.alicdn.com/tfs/TB1rmyTltfJ8KJy0FeXXXKEXXa-50-74.png',
bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCSVXXXXasXFXXXXXXXXX-100-80.png',
```

```

obj_error: '/img.alicdn.com/tfs/TB1q9yTltfJ8KJy0FeXXXKEXXa-50-74.png',
bg_back_fail: '/img.alicdn.com/tfs/TB1w2oOSFXXXb4XpXXXXXXXXXX-100-80.png',
upLang:{"cn":{
_ggk_guide: "请摁住鼠标左键，刮出两面盾牌",
_ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧",
_ggk_loading: "加载中",
_ggk_fail: ['呀，盾牌不见了<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_action_timeout: ['我等得太久啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_net_err: ['网络实在不给力<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'],
_ggk_too_fast: ['您刮得太快啦<br/>请', "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题']
}
}
}

function yourRegisterRequest(url, params){
var callbackName = ('jsonp_' + Math.random()).replace('.', '')
params += '&callback=' + callbackName
var o_scripts = document.getElementsByTagName("script")[0]
var o_s = document.createElement('script')
o_scripts.parentNode.insertBefore(o_s, o_scripts);
//您注册请求的业务回调
window[callbackName] = function(json) {
if(json.result.code == 400) {
//唤醒滑动验证
getNC().then(function(){
_nvc_nc.upLang('cn', {
_startTEXT: "请按住滑块，拖动到最右边",
_yesTEXT: "验证通过",
_error300: "哎呀，出错了，点击<a href=\"javascript:__nc.reset()\">刷新</a>再来一次",
_errorNetwork: "网络不给力，请<a href=\"javascript:__nc.reset()\">点击刷新</a>",
}))
_nvc_nc.reset()
})
} else if (json.result.code == 600) {
//唤醒刮刮卡
getSC().then(function(){} )
} else if (json.result.code == 100 || json.result.code == 200) {
//注册成功
alert("register success!")
} else if (json.result.code == 800 || json.result.code == 900) {
//直接拦截
alert("register failed!")
}
}
o_s.src = url + '?' + params
}

document.getElementById('register').onclick = function() {
var params = 'a=' + getNVCVal()
yourRegisterRequest('http://cf.aliyun.com/nvc/nvcAnalyze.jsonp', params)
}
</script>

```

```
<script src="//g.alicdn.com/sd/nvc/1.1.112/guide.js"></script>

</body>
</html>
```

## 接入注意事项

- 1.window.NVC\_Opt全局变量的声明一定要在“ //g.alicdn.com/sd/nvc/1.1.112/guide.js” 资源引入之前配置完毕
- 2.切记：yourRegisterRequest方法中维护的“ http://cf.aliyun.com/nvc/nvcAnalyze.jsonp ” 接口必须顶替为您自己的业务请求接口（现有接口属于线上Demo接口，不具备任何实际攻防能力），在这个接口调用的过程中需带上getNVCVal()获取的人机信息串，随业务请求一起带到您的业务服务端，并按照“Link:服务端接入文档” 中的指引发起“业务服务端->风控服务端” 的风险查询请求。
- 3.风险查询请求返回的结果，您可以直接返回给客户端使用，客户端可以根据风险等级选择放过、拦截、或者调用getNC()、getSC()方法在指定div上唤醒二次验证。如果您服务端维护有自己的风控策略（比如黑白名单），也可以综合您自己风控策略与人机接口二者的结果，综合在客户端决定风险处置方式。

## 参数详细说明

参数	含义
appkey	应用标示。它和scene字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
scene	场景标示。它和appkey字段一起决定了滑动验证的业务场景与后端对应使用的策略模型。您可以从console的配置里找到它正确的值，请勿乱写
isH5	是否是Html5场景，这里PC接入需填false。默认为false
popUp	决定二次验证唤醒的方式。true为浮层式，false为嵌入式
renderTo	声明二次验证需要渲染的目标元素ID
trans	业务键字段。 <b>说明：</b> 该字段可以为空。您可以按照测试指南中的描述填写该字段值，用于线上问题排查。
customWidth	当唤醒滑动验证作为二次验证时，配置滑动验证的长度
width	当唤醒刮刮卡验证作为二次验证时，配置刮刮卡的长度
height	当唤醒刮刮卡验证作为二次验证时，配置刮刮卡的高度
language	配置二次验证的语言，详见“自定义文案”部分。默认为cn(中文)

elements	两个长度的List，自定义需要刮出的两个elements的图片资源
bg_back_prepared	自定义刮动时的背景图像，平铺填充
bg_front	自定义刮动的前景（浮层），注意这里只接受base64数据流
obj_ok	自定义验证通过时显示的图标资源
bg_back_pass	自定义验证通过时的背景图像，平铺填充
obj_error	自定义验证失败或异常时显示的图标资源
bg_back_fail	自定义验证失败或异常时显示的背景图像，平铺填充
upLang	用于自定义刮刮卡文案，具体可以参见“刮刮卡交互验证前端代码集成文档”

## 方法详细说明

无痕验证维护了三个全局方法供您使用，当nvc.js资源加载完成后，这些方法您就可以直接调用了。

方法名	参数	作用
getNVCVal	无	在发送业务请求的前一刻调用此方法，并将此方法返回的信息串随业务请求带去服务端，由服务端使用此信息串调用风控服务查询人机风险
getNC	无	调用此方法会在您NVC_Opt配置项中renderTo声明的div处渲染出滑动验证
getSC	无	调用此方法会在您NVC_Opt配置项中renderTo声明的div处渲染出刮刮卡验证

## 测试指南

### (1) CF\_APP\_1复现验证码各个状态机

CF\_APP\_1这个appkey是验证码服务专门为用户测试验证码各个状态所维护的appkey。启用这个appkey后，您可以通过配置前端trans字段来直接控制验证码服务接口的返回（通过or二次验证or拦截），借此您可以直观的看到验证码各个状态的交互、样式和流程。正式上线时请务必将其替换为阿里云控制台上为您分配的appkey和scene，以免出现不必要的安全问题。

当您使用demo中提供的http://cf.aliyun.com/nvc/nvcAnalyze.jsonp 服务端接口进行测试时，CF\_APP\_1下trans字段与验证码结果对应关系如下：

trans	效果
-------	----

{ "nvcCode" :200}	无痕验证直接通过
{ "nvcCode" :400, "key1" : "code0" }	无痕验证会唤醒滑动验证，且滑动验证一定通过
{ "nvcCode" :400, "key1" : "code300" }	无痕验证会唤醒滑动验证，且滑动验证一定拦截
{ "nvcCode" :600, "key1" : "code0" }	无痕验证会唤醒刮刮卡验证，且刮刮卡验证一定通过
{ "nvcCode" :600, "key1" : "code300" }	无痕验证会唤醒刮刮卡验证，且刮刮卡验证一定拦截
{ "nvcCode" :800}	无痕验证直接拦截

## (2) 正式上线测试

正式上线之前，推荐您执行完整的功能性和兼容性测试。测试过程中有以下几点注意事项：

- 请务必将demo中的<http://cf.aliyun.com/nvc/nvcAnalyze.jsonp> 接口替换为您自己的业务接口，并按照无痕验证服务端接入文档进行后端风控接入
- 请确保您使用阿里云控制台上分配的appkey和scene进行测试，不要胡乱填写
- 请注意您当前验证码的使用场景。如果您的验证码在PCweb场景下使用，请不要用手机等移动设备进行测试
- 在测试低版本IE浏览器时，不要用IE11浏览器的兼容模式去模拟生成低版本浏览器。请用真正低版本IE浏览器进行测试

# 自定义文案

## (1) 二次弹出滑动验证文案

调用getNC()方法唤醒滑动验证时，可以在getNC().then(function{})中调用\_nvc\_nc.upLang 方法来自定义文案。自定义之后请记得调用reset()方法刷新渲染滑动验证。

```
getNC().then(function(){
  _nvc_nc.upLang('cn', {
    _startTEXT: "请按住滑块，拖动到最右边",
    _yesTEXT: "验证通过",
    _error300: "哎呀，出错了，点击<a href=\"javascript:__nc.reset()\">刷新</a>再来一次",
    _errorNetwork: "网络不给力，请<a href=\"javascript:__nc.reset()\">点击刷新</a>",
  })
  _nvc_nc.reset()
})
```

## (2) 二次弹出刮刮卡验证文案

调用getSC()方法唤醒刮刮卡验证时，自定义文案可以直接在NVC\_Opt配置项中配置upLang来修改。

```
upLang:{"cn":{
```

```
_ggk_guide: "请摁住鼠标左键，刮出两面盾牌", //初始化完成
_ggk_success: "恭喜您成功刮出盾牌<br/>继续下一步操作吧", //验证成功
_ggk_loading: "加载中", //加载过程
_ggk_fail: ["呀，盾牌不见了<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'], //验证失败
_ggk_action_timeout: ["我等得太久啦<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'], //操作超时
_ggk_net_err: ["网络实在不给力<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'], //加载失败
_ggk_too_fast: ["您刮得太快啦<br/>请", "javascript:noCaptcha.reset()", '再来一次', '或',
"http://survey.taobao.com/survey/QgzQDdDd?token=%TOKEN", '反馈问题'], //刮动过快
}
}
```

## 人机验证-使用说明

## 智能验证WEB/HTML5使用说明

### 简介

智能验证以一个button的形式提供给接入方集成人机识别的能力，大部分用户只需要一次点击交互即可完成原先复杂的人机验证码操作。

### 产品展示





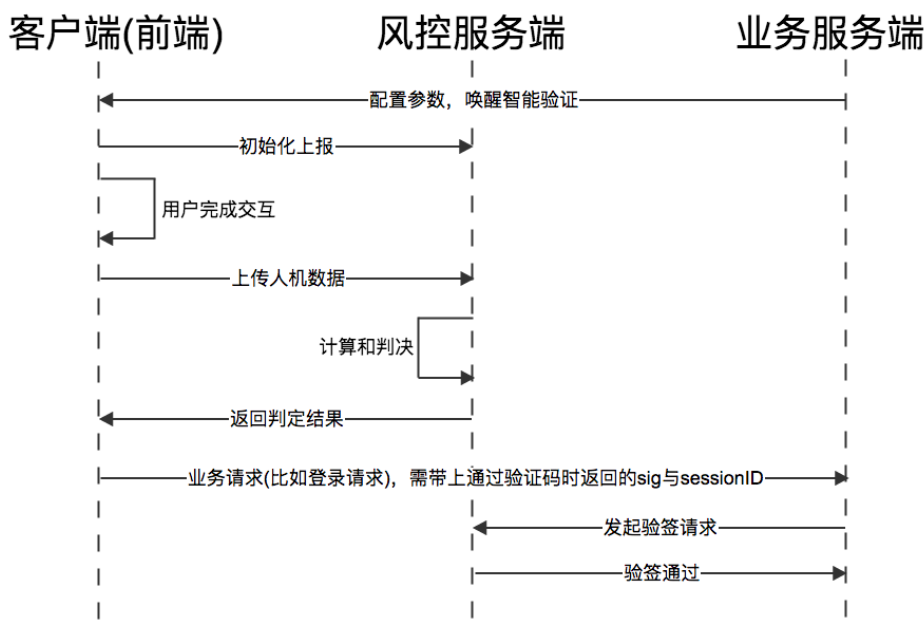




点击按钮开始智能验证

登 录

系统交互流程



接入流程

- 1.进入阿里云控制台-云盾-数据风控-验证码服务，进入“配置管理”标签。
- 2.如果您已经在您需要接入智能验证的场景下创建过智能验证验证的配置，请复用以前的配置和接入代码；否则请点击“新增配置”。

3.请配置您的“服务名称”、“高峰期PV”、“使用场景”、“应用类型”，并在下方“验证方式”处选择“智能验证”，点击下一步。

\*注意\*：这里您的配置项一旦创建即写入线上配置，不可修改，请谨慎配置。

4.在“系统代码集成&测试”流程中保存您需要的前端代码与服务端代码。这里您也可以参照“前端代码集成”与“服务端代码集成”部分文档来进行接入。请妥善保存此处的代码，并点击下一步。

5.接入完成。您可以在“配置管理”处点击“自定义样式”来可视化的自定义一些您的配置，也可以根据“智能验证前端代码集成文档”来自定义配置。

6.我们在控制台（控制台详情介绍：[点击查看](#)）上为您提供了报表、监控和简单的预警功能，请按需使用。

## 刮刮卡WEB/HTML5使用说明

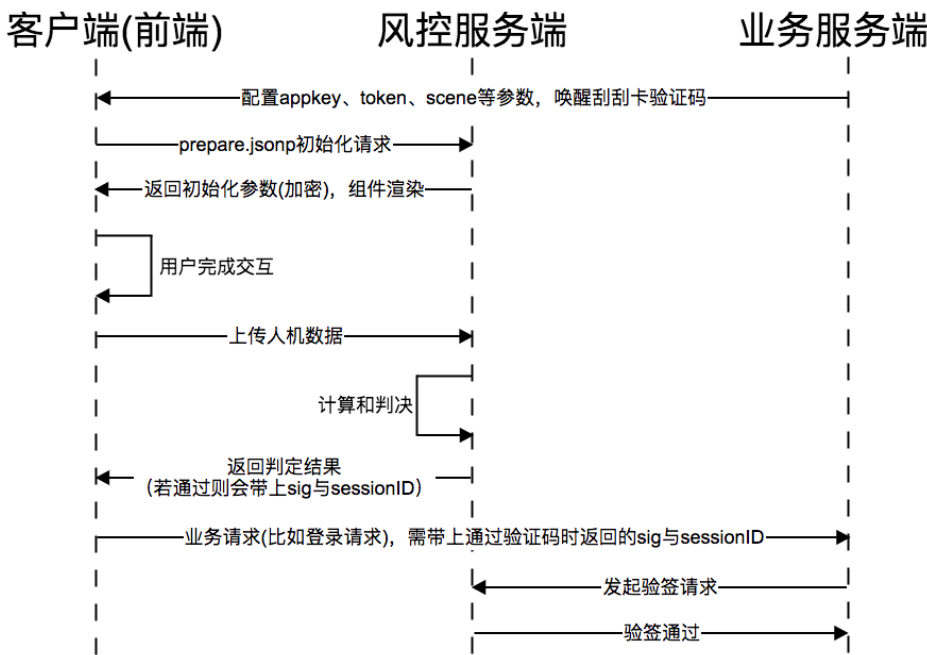
### 简介

刮刮卡交互验证码(Scratchcard-Captcha)是阿里集团推出的第二代行为验证码产品，它采用先进的风险分析引擎外加机器学习模型来区分人类和机器。刮刮卡交互验证码需要用户与页面上刮刮卡UI组件进行交互，刮出特定内容即可通过验证，用户无需思考即可通过人机识别（图灵测试）的挑战。

### 产品展示



系统交互流程



## 接入流程

- 1.进入阿里云控制台-云盾-数据风控-验证码服务，进入“配置管理”标签。
  - 2.如果您已经在您需要接入刮刮卡验证的场景下创建过滑动验证的配置，请复用以前的配置和接入代码；否则请点击“新增配置”。
  - 3.请配置您的“服务名称”、“高峰期PV”、“使用场景”、“应用类型”，并在下方“验证方式”处选择“刮刮卡验证”，点击下一步。
- \*注意\***：这里您的配置项一旦创建即写入线上配置，不可修改，请谨慎配置。
- 4.在“系统代码集成&测试”流程中保存您需要的前端代码与服务端代码。这里您也可以参照“前端代码集成”与“服务端代码集成”部分文档来进行接入。请妥善保存此处的代码，并点击下一步。
  - 5.接入完成。您可以在“配置管理”处点击“自定义样式”来可视化的自定义一些您的配置，也可以根据“刮刮卡验证前端代码集成文档”来自定义配置。
  - 6.我们在控制台（控制台详情介绍：[点击查看](#)）上为您提供了报表、监控和简单的预警功能，请按需使用。

## 无痕验证WEB/HTML5使用说明

### 简介

无痕验证是阿里巴巴提供的新一套人机解决方案。由无痕组件负责收集和统一调度下游验证码产品实现综合人机对抗。

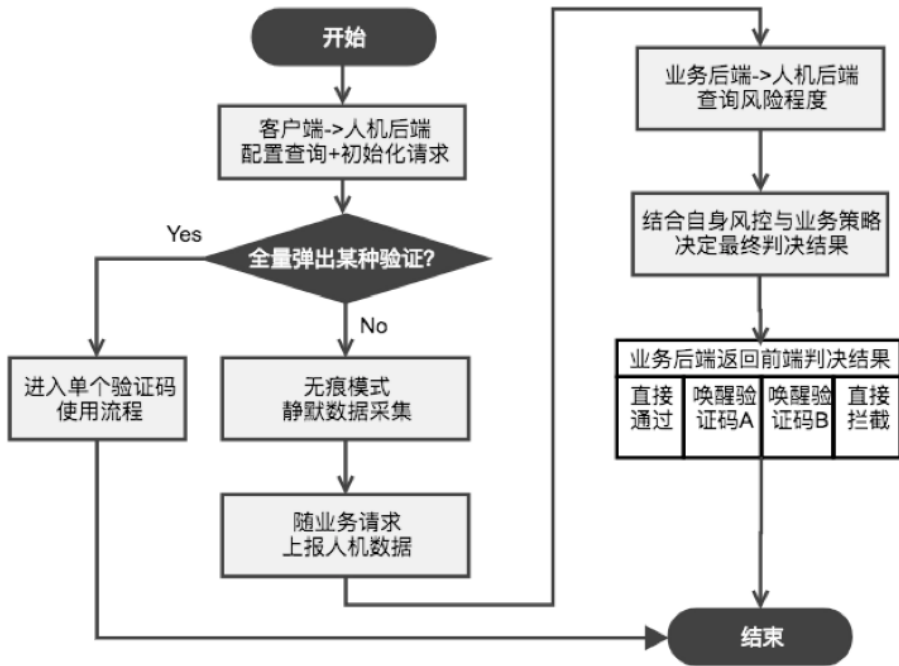
用户接入后完全静默采集，对大部分正常用户0打扰，对有风险用户会根据风险等级返回相应的下游验证码产品唤醒建议。

无痕验证采用服务端-服务端的风险查询机制，业务方在有风控能力的情况下可以很轻松的将无痕验证人机识别的能力并联到自己的业务风控上。

### 产品展示

登 录

系统交互流程



接入流程

- 1.进入阿里云控制台-云盾-数据风控-验证码服务，进入“配置管理”标签。
  - 2.如果您已经在您需要接入智能验证的场景下创建过无痕验证验证的配置，请复用以前的配置和接入代码；否则请点击“新增配置”。
  - 3.请配置您的“服务名称”、“高峰期PV”、“使用场景”、“应用类型”，并在下方“验证方式”处选择“无痕验证”，点击下一步。
- \*注意\*：这里您的配置项一旦创建即写入线上配置，不可修改，请谨慎配置。

- 4.在“系统代码集成&测试”流程中保存您需要的前端代码与服务端代码。这里您也可以参照“前端代码集成”与“服务端代码集成”部分文档来进行接入。请妥善保存此处的代码，并点击下一步。
- 5.接入完成。您可以在“配置管理”处点击“自定义样式”来可视化的自定义一些您的配置，也可以根据“无痕验证前端代码集成文档”来自定义配置。
- 6.我们在控制台（控制台详情介绍：[点击查看](#)）上为您提供了报表、监控和简单的预警功能，请按需使用。

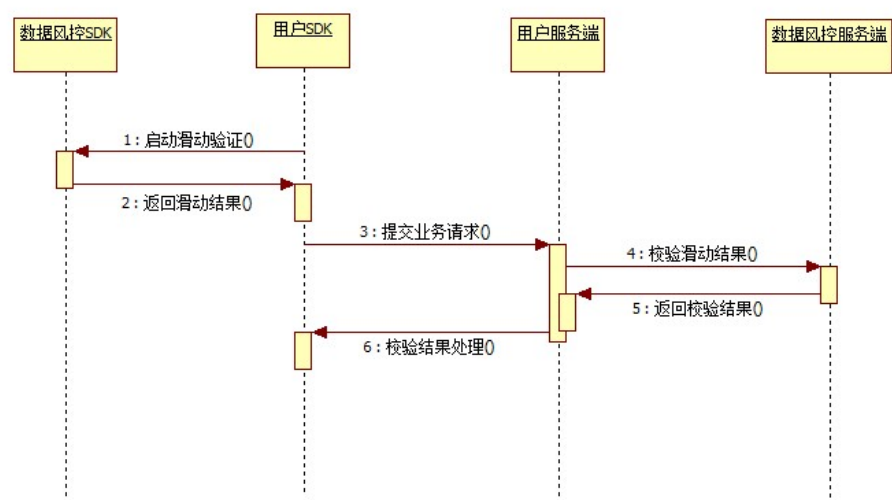
## 投篮验证Android/iOS使用说明

**滑动验证**，通过生物特征判定操作计算机的是人还是机器，从而取代传统验证方式。Android、iOS端滑动验证组件展现形式：

**Android/iOS：**



## 系统间交互流程



- 1.启动滑动验证：用户APP调用风控SDK启动风险验证。用户在页面操作滑动验证：
- 正常用户：滑动直接通过；
  - 有风险用户：滑动后失败，直接返回初始页面，提示重新滑动；
- 2.提交业务请求：用户提交业务请求如注册、登陆请求时，需要将验证码参数传递给服务端，验证码参数包含：
- session\_id：验证会话id
- 3.校验验证结果：用户服务端调用验证码服务api，获得校验结果。该步骤需要放在业务请求处理之前。
- 4.校验结果处理：根据返回的校验结果，进行处理。对于不同校验结果，建议处理方案：
- 校验成功：继续做业务请求处理，如注册、登陆处理；
  - 校验失败：建议直接返回业务请求失败，让操作者重新再来一次。

接入流程

- 1.第一步，进入数据风控控制台：
- 未开通服务，开通服务；
  - 已开通服务，进入服务管理，选择滑动验证服务，点击【创建】：





2.第二步，选择应用类型（Android、iOS）、使用场景、输入高峰期PV，点击【下一步】：



3.第三步，根据系统集成代码的操作步骤，下载SDK，集成到APP中；服务端调用对应api，详细操作参见下方；系统集成。集成完成，点击【下一步】：



4.第四步，系统集成完毕，发布对应的服务：



## 系统集成

### Android SDK集成配置

#### 1.生成SDK

出于对应用数据安全考虑，数据风控生成的SDK会与应用强绑定。如果此前使用Debug版应用 生成SDK需要在应用发布前使用Release版应用重新上传，并替换原有SDK。

- 上传Release版APK；
- 点击 “生成SDK” 按钮，生成SDK；
- 生成完SDK后，SDK自动下载到本地；

#### 2.导入SDK

##### 2.1 导入前准备

数据风控SDK在阿里巴巴的很多开放的SDK中也有包含，如果你的APK中同时引入了这些SDK，则在集成聚安

全SDK之前需要将这些SDK ( TAE , 支付宝等)中包含的安全组件 ( .jar、.aar文件 )、图片文件 ( YW\_1222.JPG ) 全部删除掉。

## 2.2 导入SDK

根据使用方式导入SDK，AndroidStudio使用aar方式导入，Eclipse使用jar和so方式导入；

- 导入aar，如图把所有的aar都复制到项目的libs目录下，然后在该Module的build.gradle中增加如图



```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.+'
    compile name: 'NoCaptchaSDK-external-release-5.1.17', ext: 'aar'
    compile name: 'SecurityBodySDK-external-release-5.1.25', ext: 'aar'
    compile name: 'SecurityGuardSDK-external-release-5.1.81', ext: 'aar'
    compile name: 'verificationsdklib', ext: 'aar'
}
```

- 导入jar包和so，如图把SDK中的所有的jar包和so文件都复制到工程的libs目录下：



注意：

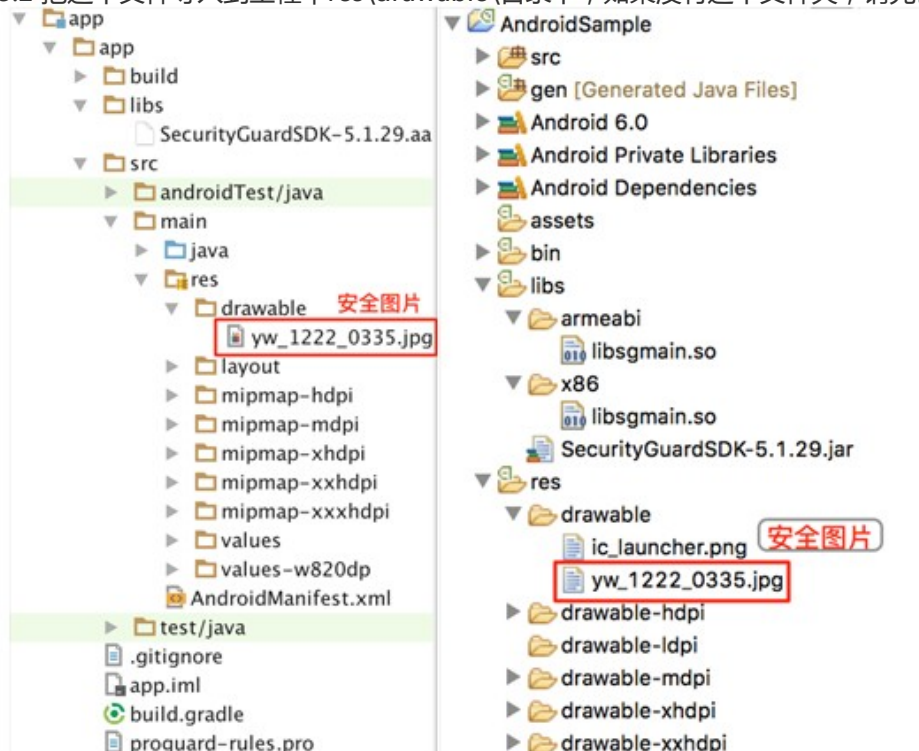
- a. 目前只提供两种架构下的so文件，armeabi是针对arm架构编译的包，x86是针对x86架构编译的包。应用程序在不同cpu架构的机型里会选择相应的so文件加载。
- b. 如果libs下有armeabi-v7a文件夹的话，需要将armeabi中对应的so复制一份到armeabi-v7a文件夹下。
- c. 如果libs下有arm64-v8a或x86\_64文件夹的话，需要将arm64-v8a文件夹删除掉。
- d. 如果在想在x86或者模拟器上运行你的程序，必须导入x86架构的so，但是可以在应用发布时去掉x86目录下的so。
- e. 最后目录显示如下：

```
NoCaptchaSDK-5.1.16.jar
armeabi
SecurityBodySDK-5.1.19.jar
verificationdklib.aar
SecurityGuardSDK-5.1.58.jar
x86
```

### 3.导入图片

3.1 解压第1点中生成的SDK，获得文件：yw1222.jpg；

3.2 把这个文件导入到工程中res\drawable\目录下，如果没有这个文件夹，请先创建，如下图：



左图为AndroidStudio中安全加密图片位置，右图为Eclipse中安全加密图片位置；

3.3 如果开启混淆要检查发布包335大小不为0，shrinkResources true会导致yw\_1222\_0335以及yw\_1222图片为0

```
release {
    minifyEnabled true // 是否混淆
    shrinkResources true // <<<会导致335或者122图片为0
    proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
}
```

#### 解决方案一

放弃资源压缩

#### 解决方案二

参考google 关于shrink resource

新建 res/raw/keep.xml后加入如下内容:

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns:tools="http://schemas.android.com/tools"
tools:keep="@drawable/yw_1222_0335, @drawable/yw_1222"/>
```

- 保留文件规则简单介绍，资源文件相对路径加上图片文件名(不需要扩展名)；
- 执行 ./gradlew clean assembleRelease -info|grep "Skipped unused resource" 观察是否安全图片给压缩,同时检查解压后文件是否为0。

## 4.android studio修改项目文件

### 4.1 修改应用的工程根目录build.gradle文件：

```
allprojects {
    repositories {
        jcenter()
        flatDir { //<-----添加三行
            dirs 'libs' //<-----
        } //<-----
    }
}
```

### 4.2 修改application子工程的build.gradle文件

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile(name:'verificationsdklib', ext:'aar')
}
```

### 4.3 修改application子工程的build.gradle文件，增加jnilib

```
sourceSets {
    main {
        jniLibs.srcDirs = ['libs']
    }
}
```

### 4.4 AndroidManifest.xml 加入验证SDK入口Activity声明

```
<activity android:name="com.alibaba.verificationsdk.ui.VerifyActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.NoTitleBar"
    android:windowSoftInputMode="adjustResize" >
    </activity>
```

## 5.Eclipse修改项目文件5.1 添加权限信息

- 如果是AndroidStudio项目，则不需要在项目中额外配置权限，因为在aar中我们自己已经声明了权限；
- 如果是Eclipse项目，需要在AndroidManifest.xml文件中添加下列权限配置：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
```

## 5.2 AndroidManifest.xml 加入验证SDK入口Activity声明

```
<activity android:name="com.alibaba.verificationsdk.ui.VerifyActivity"
    android:screenOrientation="portrait"
    android:theme="@android:style/Theme.NoTitleBar"
    android:windowSoftInputMode="adjustResize" >
</activity>
```

## 6.关于混淆

如果设置资源压缩 `shrinkResources true`，参考导入图片的处理，防止安全图片被压缩为0字节，配置 `proguard-rules.pro`：

```
-keep class com.taobao.securityjni.**{*;}
-keep class com.taobao.wireless.security.**{*;}
-keep class com.ut.secbody.**{*;}
-keep class com.taobao.dp.**{*;}
-keep class com.alibaba.wireless.security.**{*;}
-keep class com.alibaba.verificationsdk.**{*;}
-keep interface com.alibaba.verificationsdk.ui.IActivityCallback
```

## 7.SDK API

### 7.1 SDK初始化：

- 初始化负责完成整个数据风控安全组件的全局初始化。初始化是线程安全的，初始化调用只需要进行一次，无需重复调用；
- 查看接口详情

### 7.2 启动风险验证

- 使用场景：在需要使用验证码的场景，如注册、登陆、活动页面，可以直接启动验证码进行验证；
- 查看接口详情

## iOS SDK集成配置

### 1.生成SDK

出于对应用数据安全考虑，数据风控生成的SDK会与应用强绑定。如果此前使用Debug版应用 生成SDK需要在应用发布前使用Release版应用重新上传，并替换原有SDK。

- 上传Release版ipa；
- 点击“生成SDK”按钮，生成SDK；
- 生成完SDK后，SDK自动下载到本地；

## 2. 导入SDK

### 2.1 导入Framework

a. 把SDK中的framework文件添加到项目目录中

```
MSAuthSDK.framework SecurityGuardSDK.framework  
SGMain.framework`SGNoCaptcha.framework`SGSecurityBody.framework
```

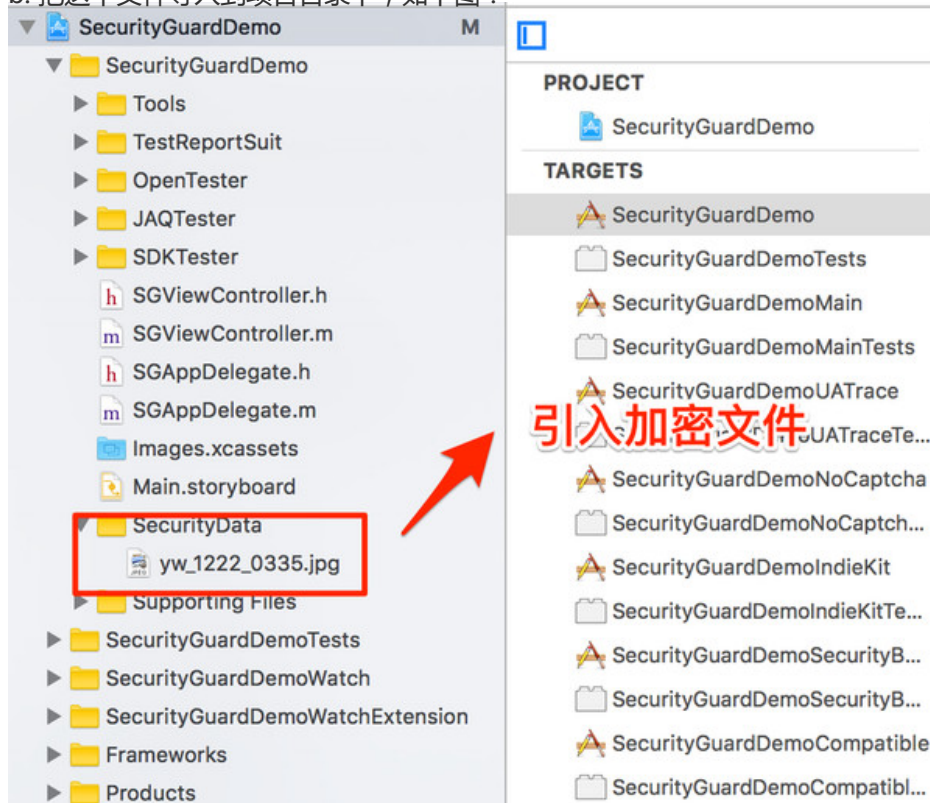
b. 将MSADefaultImages.bundle和MSADefaultLocale.bundle加入资源中

c. 如果sdk中带有xib，需要将所有xib加入资源中

### 2.2 导入图片

a. 解压第1点中生成的SDK，获得文件：yw1222 \*.jpg；

b. 把这个文件导入到项目目录下，如下图：



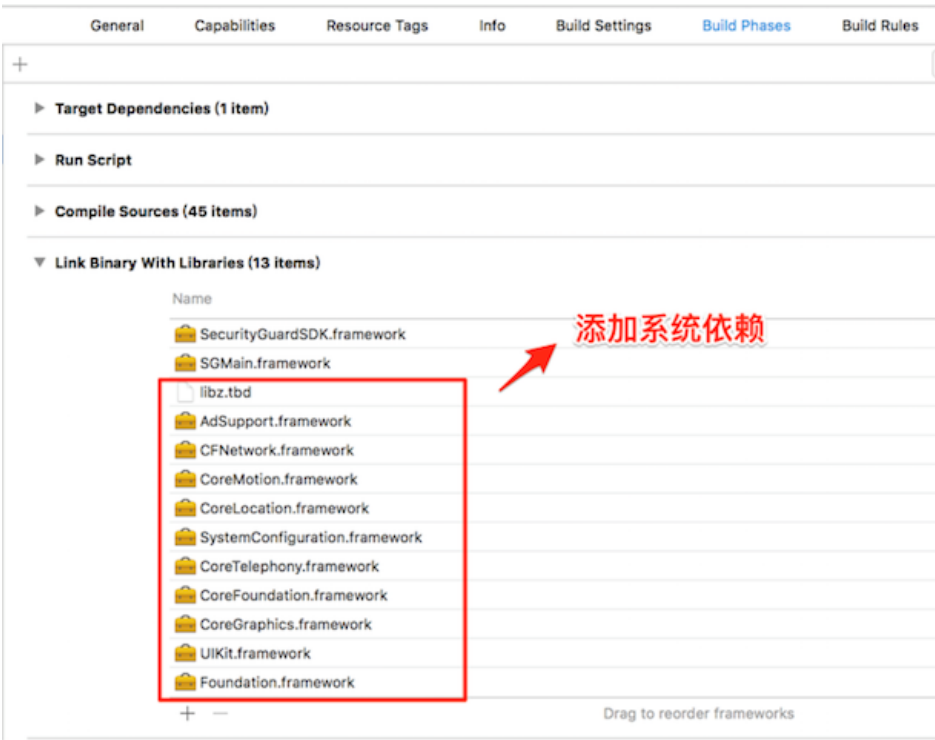
### 2.3 依赖 CocoaPods:

不使用pod可以直接使用sdk内压缩包：

```
pod 'SVProgressHUD', '~> 1.1'  
pod 'SSZipArchive', '~> 1.1'
```

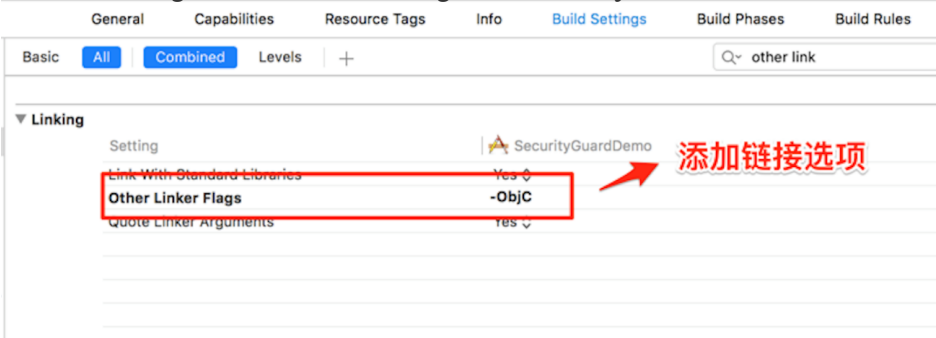
### 2.4 添加系统依赖库

在项目中添加其他依赖的framework，如下图：



2.5 其他项目配置

- 在Build Setting中的Other Linker Flags中添加 “-ObjC” 选项，如下图：



- 在info.plist中设置，开放http请求：

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key> <true/>
</dict>
```

3.SDK API

3.1 启动风险验证

服务端API调用

1. 下载对应语言的SDK，SDK下载请进入数据风控控制台；



2. 将SDK加载到工程中；
3. 参考下方的代码，开发第一步页面请求的处理类（如java的Action,Controller,Servlet等）；注意填入自己的阿里云accesskey和secret；
4. 参考示例代码，对运行结果进行处理；请注意对服务端出现的错误进行兼容处理。
5. API详情：验证码服务Android/iOS接口

## 滑动验证WEB/HTML5使用说明

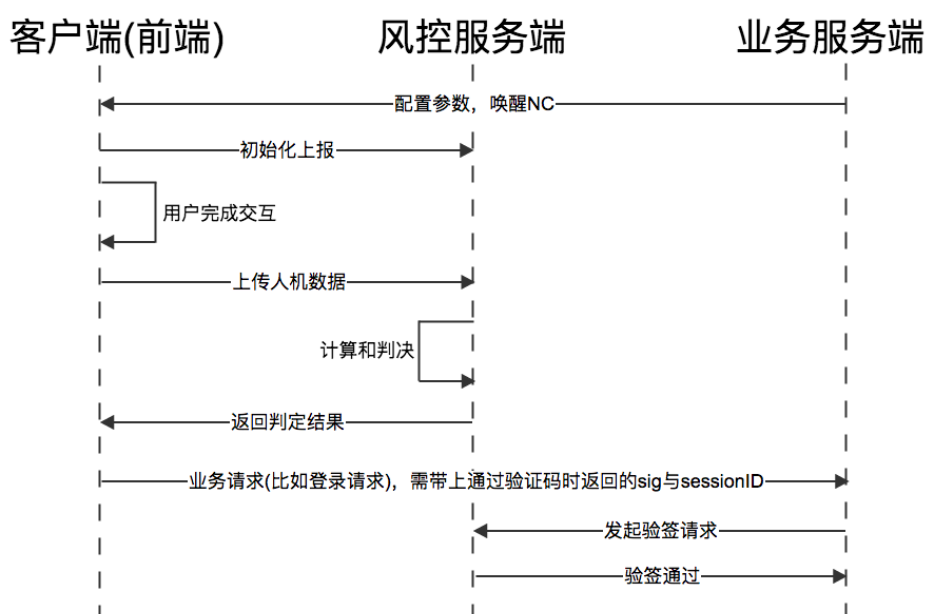
### 简介

No-Captcha滑动验证，简称为NC。是阿里集团突破传统验证码的最新人机识别产品，它采用先进的风险分析引擎来区分人类和机器人。用户通过类似滑动解锁的形式进行人机判断，用户无需思考即可通过人机识别（图灵测试）的挑战。

### 产品展示



### 系统交互流程



## 接入流程

- 1.进入阿里云控制台-云盾-数据风控-验证码服务，进入“配置管理”标签。
- 2.如果您已经在您需要接入滑动验证的场景下创建过滑动验证的配置，请复用以前的配置和接入代码；否则请点击“新增配置”。
- 3.请配置您的“服务名称”、“高峰期PV”、“使用场景”、“应用类型”，并在下方“验证方式”处选择“滑动验证”，点击下一步。
- \*注意\*：这里您的配置项一旦创建即写入线上配置，不可修改，请谨慎配置。
- 4.在“系统代码集成&测试”流程中保存您需要的前端代码与服务端代码。这里您也可以参照“前端代码集成”与“服务端代码集成”部分文档来进行接入。请妥善保存此处的代码，并点击下一步。
- 5.接入完成。您可以在“配置管理”处点击“自定义样式”来可视化的自定义一些您的配置，也可以根据“滑动验证前端代码集成文档”来自定义配置。
- 6.我们在控制台（控制台详情介绍：[点击查看](#)）上为您提供了报表、监控和简单的预警功能，请按需使用。

## 人机验证-服务端代码集成

## API参考

# 人机验证服务Android/iOS API

## 描述

Android和iOS验证码结果校验接口，确定前端页面获取到的签名串是否验证码服务端颁发。

## 公共请求参数

参见：公共请求参数

## 业务请求参数

名称	类型	是否必须	描述
Session	String	是	二次验证获取验证检查结果所需的seesionId

## 请求示例

```
http://jaq.aliyuncs.com/?
Action= AfsCheck
&Session=xxxxx
& ... (加上公共入参)
```

## 响应结果

成功响应结果：

```
json :
{
  "data": true
}
```

失败响应结果：

```
json :
{
  "Message":"invalid parameter.",
  "RequestId":"A649BA1E-76CF-4CD7-82D7-BD835533C96F", "HostId":"jaq.aliyuncs.com",
  "Code":"400"
}
```

特定错误码：

错误码	值	含义
errorCode	400	参数错误
errorCode	500	系统内部错误

## 人机验证服务web/html5 API

### SDK接入指引

参见：  
java sdk接入指引  
python sdk接入指引  
php sdk接入指引  
.net sdk接入指引

### 描述

PC WEB或移动端HTML5验证码结果校验接口，确定前端页面获取到的签名串是否验证码服务端颁发。

### 公共请求参数

参见：公共请求参数

### 业务请求参数

名称	类型	是否必须	描述
Token	String	是	请求唯一标识
Sig	String	是	签名串
SessionId	String	是	会话ID
Scene	String	否	报表统计中用到，务必传
AppKey	String	是	appkey
RemoteIp	String	是	客户的用户端 RemoteIp

## 请求示例

```
http://afs.aliyuncs.com/?
Action=AuthenticateSig
&SessionId=13211111111
&Scene=xxx
&Token=xxx
&Sig=1
&AppKey=xxx
&RemoteIp=xxx
&公共请求参数
```

## 响应结果

成功响应结果示例（JSON格式）

```
{
  code = "200";
  success = true;
  data {
    code=100
  }
}
```

失败响应结果示例（JSON格式）

```
{
  code = "200";
  success = true;
  data {
    code=900
  }
}
```

特定错误码：

错误码	值	含义
errorCode	400	参数错误
errorCode	500	系统内部错误

## API概览

本手册将详细介绍数据风控服务API的功能、使用方法及详细参数的设定。

## iOS SDK API

iOS SDK API详情：[点击查看](#)

API	描述
获取数据风控wtoken	在需要使用数据风控的场景，如注册、登陆等，在进行业务逻辑处理前获得该token，服务端通过该token查询风险结果
启动验证	启动验证码
验证结果回调	关闭验证码

## Android SDK API

Android SDK API详情：[点击查看](#)

API	描述
初始化	完成整个数据风控组件的全局初始化
获取数据风控wtoken	在需要使用数据风控的场景，如注册、登陆等，在进行业务逻辑处理前获得该token，服务端通过该token查询风险结果
启动验证	启动验证码
验证结果回调	验证过程中终止或者验证结束时触发
验证界面回调	在用户取消验证情况下调用

## 验证码服务API

API	描述
AfsCheck	验证码服务web/html5接口
AfsAppCheck	验证码服务Android/iOS接口

## 相关文档

- [API资源导航](#)
- [API Explorer](#)
- [API错误中心](#)

# 公共参数

## 公共请求参数

### 相关参数说明

公共请求参数是指每个接口都需要使用到的请求参数。

名称	类型	是否必须	描述
Format	String	否	返回值的类型，支持JSON与XML。默认为XML
Version	String	是	API版本号，为日期形式：YYYY-MM-DD 取值：2016-11-23
AccessKeyId	String	是	阿里云颁发给用户的访问服务所用的密钥ID
Signature	String	是	签名结果串，关于签名的计算方法，请参见签名机制。
SignatureMethod	String	是	签名方式，目前支持HMAC-SHA1
Timestamp	String	是	请求的时间戳。日期格式按照ISO8601标准表示，并需要使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ;例如，2013-08-15T12:00:00Z（为北京时间2013年8月15日20点0分0秒）
SignatureVersion	String	是	签名算法版本，目前版本是1.0
SignatureNonce	String	是	唯一随机数，用于防止网络重放攻击。用户在不同请求间要使用不同的随机数值

## 示例

```
https://cf.aliyuncs.com/  
?Format=xml  
&Version=2014-08-15  
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dgI%3D  
&SignatureMethod=HMAC-SHA1  
&SignatureNonce=15215528852396  
&SignatureVersion=1.0  
&AccessKeyId=key-test  
&OwnerId=12345678  
&Timestamp=2014-10-10T12:00:00Z
```

## 公共返回参数

用户发送的每次接口调用请求，无论成功与否，系统都会返回一个唯一识别码RequestId给用户。

## 示例

```
<?xml version="1.0" encoding="utf-8"?>  
<!--结果的根结点-->  
<接口名称+Response>  
<!--返回请求标签-->  
<RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>  
<!--返回结果数据-->  
</接口名称+Response>
```

## 返回结果

调用API服务后返回数据采用统一格式，返回的HTTP状态码为2xx，代表调用成功；返回4xx或5xx的HTTP状态码代表调用失败。调用成功返回的数据格式主要有XML和JSON两种，外部系统可以在请求时传入参数来制定返回的数据格式，默认为XML格式。本文档中的返回示例为了便于用户查看，做了格式化处理，实际返回结果是没有进行换行、缩进等处理的。

## 成功结果

XML返回结果包括请求是否成功信息和具体的业务数据。示例如下：

```
<?xml version="1.0" encoding="utf-8"?>  
<!--结果的根结点-->  
<接口名称+Response>  
<!--返回请求标签-->  
<RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>  
<!--返回结果数据-->  
</接口名称+Response>
```



JSON示例：

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
  /* 返回结果数据 */
}
```

## 错误结果

调用接口出错后，将不会返回结果数据。当调用出错时，HTTP请求返回一个4xx或5xx的HTTP状态码。返回的消息体中是具体的错误代码及错误信息。另外还包含一个全局唯一的请求ID：RequestId和一个您该次请求访问的站点ID：HostId。在调用方找不到错误原因，可以联系阿里云客服，并提供该HostId和RequestId，以便我们尽快帮您解决问题。

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <HostId>rds.aliyuncs.com</HostId>
  <Code>UnsupportedOperation</Code>
  <Message>The specified action is not supported.</Message>
</Error>
```

JSON示例：

```
{
  "RequestId": "7463B73D-35CC-4D19-A010-6B8D65D242EF",
  "HostId": "rds.aliyuncs.com",
  "Code": "UnsupportedOperation",
  "Message": "The specified action is not supported."
}
```

## 请求结构

### 服务地址

反欺诈API的服务接入地址为：cf.aliyuncs.com

### 通信协议

为了获得更高的安全性，反欺诈服务仅支持通过HTTPS通道进行请求通信。

## 请求方法

支持HTTPS GET方法发送请求，这种方式下请求参数需要包含在请求的URL中。

## 请求参数

每个请求都需要指定要执行的操作，即Action参数（例如Authenticate），以及每个操作都需要包含的公共请求参数和指定操作所特有的请求参数。

## 字符编码

请求及返回结果都使用UTF-8字符集进行编码。

# Android SDK API

## 数据风控-Android SDK API

### 验证类型定义

```
/**
 * 当前支持的验证类型：目前支持的类型有滑动验证，即将会有更多验证方式加入
 */
import com.alibaba.verificationsdk.ui.VerifyType;

public enum VerifyType {
    NOCAPTCHA(3);
    private int value;
    private VerifyType(int value) {
        this.value = value;
    }
    public int getValue() {
        return value;
    }
}
```

### 启动验证

```
import com.alibaba.verificationsdk.ui.VerifyActivity;
/**
```

```
* 唤起验证UI
*
* @param ctx 传入调用Activity的上下文
* @param verifyType 目前滑动验证
* @param IActivityCallback 验证界面回调接口，
* 由应用自己实现
*/
public static void startSimpleVerifyUI(Context ctx, //
VerifyType verifyType,
IActivityCallback callback)
```

## 验证界面取消回调

在用户取消验证情况下调用

```
public interface IActivityCallback {
onNotifyBackPressed()
}
```

## 风险验证结果回调

验证过程中终止或者验证结束时触发

```
import com.alibaba.verificationsdk.ui.VerifyActivity;
/**
* 风险验证结果回调
*
* @param retInt 验证返回状态码
* @param detail 返回详细信息，包含错误信息，二次验证用sessionId
*/
public interface IActivityCallback {
void onResult(int retInt, Map<String, String> code);
}
```

retInt验证返回状态码:

值	含义
VERIFY_SUCC	验证成功
VERIFY_FAILED	验证失败

detail返回值详情：

值	含义
errorMsg	失败时查详细的错误 / 失败信息
sessionID	验证成功 / 失败返回sessionID做二次验证，如果异常会返回null
errorCode	失败类型

关于detail中errorCode详细定义：

值	含义
ERROR_FROM_SERVER	表示服务端出错
ERROR_FROM_CLIENT	表示网络等问题，返回数据出错
ERROR_FROM_BUSSINESS	表示具体业务出错

## 关闭验证UI的接口

```
VerifyActivity.finishVerifyUI();
```

## 使用验证码示例代码

### Step 0: 初始化安全组件

```
private SecurityVerification securityVerification = null;
Context context = this.getApplicationContext();
try {
    SecurityInit.Initialize(context);
    securityVerification = new SecurityVerification(context);
} catch (JAQException e) {
    e.printStackTrace();
}
```

### Step 1: 风险验证

```
VerifyActivity.startSimpleVerifyUI(LoginActivity.this,
VerifyActivity.VerifyType.NOCAPTCHA, new VerifyActivity.IActivityCallback() {
@Override
public void onNotifyBackPressed() {

}

@Override
public void onResult(int retInt, Map<String, String> code) {
switch (retInt) {
case VerifyActivity.VERIFY_SUCC:
Log.e(LOG_TAG,code.get("sessionID"));
break;
case VerifyActivity.VERIFY_FAILED:
Log.e(LOG_TAG,code.get("errorCode"));
Log.e(LOG_TAG,code.get("errorMsg"));
break;
default:
break;
}
}
});
```

### Step 3: sessionid 验证

具体调用参看 [API概览-验证码服务API](#)

## IOS SDK API

### 数据风控-iOS SDK API

#### 验证类型定义

MSAuthDefines.h

```
/**
 * 当前支持的验证类型：目前支持的类型有滑动验证，即将会有更多验证方式加入
 */
typedef enum : NSUInteger {
    MSAuthTypeSlide
} MSAuthType;
```

#### 启动验证

```
/**
 * 验证VC的工厂方法，使用风险验证用户直接调用这个接口
 * @param type 验证类型。目前支持的类型有滑动验证，即将会有更多验证方式加入
 * @param language 需要展示的语言，如：zh_CN 简体中文, en 英文。nil为当前环境。如果语言包不存在显示英文
 * @param delegate 处理回调结果的代理
 * @return 返回一个实例VC
 */

+ (UIViewController *)simpleVerifyWithType:(MSAuthType)type
language:(NSString*)lang
Delegate:(id<MSAuthProtocol>)delegate;
```

#### 验证结果回调

用户通过回调函数关闭controller，sdk内部不会关闭此控件。

```
typedef enum {
    VERIFY_REUSLT_OK = 0,
    VERIFY_REUSLT_FAIL = 1
}t_verify_reuslt
```

```
@protocol MSAuthProtocol <NSObject>

/**
 * 验证结果的回调
 *
 * @param error 验证错误返回码， nil表示正确 非空为验证失败
 * @param sessionId 聚安全服务器下发二次验证sessionId,
 * 三方服务器利用token获得验证结果
 */
- (void)verifyDidFinishedWithResult:(t_verify_reuslt)code
Error:(NSError *)error
SessionId:(NSString *)sessionId;
@end
```

参数error对象 ErrorDomain详细定义：

值	含义
AliAuthGeneric	一般性错误来自网络异常
AliAuthServer	网关通信错误
AliAuthServiceNc	滑动业务错误
AliAuthService	验证业务错误

## 使用验证码示例代码

### Step 0: 引入头文件

```
#import <MSAuthSDK/MSAuthSDK.h>
```

### Step 1: 调用工厂类生成所需二次验证 viewController

```
UIViewController *vc = [MSAuthVCFactory simapleVerifyWithType:服务器回传的验证类型
language:@"zh_CN"
Delegate:self];
[self.navigationController pushViewController:vc animated:YES];
```

### Step 2: 实现delegate回调

```
@interface ViewController () <MSAuthProtocol>

...

#pragma mark - MSAuthProtocol methods

- (void)verifyDidFinishedWithError:(NSError *)error SessionId:(NSString *)sessionId {

dispatch_async(dispatch_get_main_queue(), ^{
if (error) {
```

```
NSLog(@"验证失败 %@", error);
} else {
NSLog(@"验证通过 %@", sessionId);
}
[self.navigationController popViewControllerAnimated:YES];
//将sessionId传到经过app服务器做二次验证
});
}
```

### Step 3:sessionId 验证

具体调用参看 [API概览-验证码服务API](#)

## 错误处理

1.如看到以下报错：

App Transport Security has blocked a cleartext HTTP (http://) resource load since it is insecure.

Temporary exceptions can be configured via your app' s Info.plist file.

SG ERROR: 1210

你需要在info.plist中设置

```
<key>NSAppTransportSecurity</key>
<dict>
<key>NSAllowsArbitraryLoads</key> <true/>
</dict>
```

## 签名机制

数据风控服务会对每个访问的请求进行身份验证，通过使用Access Key ID和Access Key Secret进行对称加密的方法来验证请求的发送者身份。Access Key ID和Access Key Secret由阿里云官方颁发给访问者（可以通过阿里云官方网站申请和管理），其中Access Key ID用于标识访问者的身份；Access Key Secret是用于加密签名字符串和服务端验证签名字符串的密钥，必须严格保密，只有阿里云和用户知道。

## 签名步骤

### 1. 使用请求参数构造规范化的请求字符串（Canonicalized Query String）

a. 按照参数名称的字典顺序对请求中所有的请求参数（包括文档中描述的“公共请求参数”和给定了的请求接口的自定义参数，但不能包括“公共请求参数”中提到Signature参数本身）进行排序；注：当使用GET方法提

交请求时，这些参数就是请求URI中的参数部分（即URI中“?”之后由“&”连接的部分）；b. 对每个请求参数的名称和值分别进行URL编码（仅支持UTF-8字符集）' ' '。URL编码的编码规则是：

- 对于字符 A-Z、a-z、0-9以及字符“-”、“\_”、“.”、“~”不编码；
- 对于其他字符编码成“%XY”的格式，其中XY是字符对应ASCII码的16进制表示。比如英文的双引号（"）对应的编码就是%22；
- 对于扩展的UTF-8字符，编码成“%XY%ZA...”的格式；
- 需要说明的是英文空格（ ）要被编码是%20，而不是加号（+）。注意：一般支持URL编码的库（比如Java中的java.net.URLEncoder）都是按照“application/x-www-form-urlencoded”的MIME类型的规则进行编码的。实现时可以直接使用这类方式进行编码，把编码后的字符串中加号（+）替换成%20、星号（\*）替换成%2A、%7E替换回波浪号（~），即可得到上述规则描述的编码字符串。

c. 将编码后的参数名称和值使用英文等号（=）连接，得到若干参数对；d. 将参数对按照原参数名称字典序排序结果使用&符号依次连接，即得到规范化请求字符串。

## 2. 使用上一步构造的规范化字符串按照下面的规则构造用于计算签名的字符串

生成方式如下：

```
StringToSign=
HTTPMethod + "&" +
percentEncode("/") + "&" +
percentEncode(CanonicalizedQueryString)
```

- 其中HTTPMethod是提交请求用的HTTP方法，比如GET。
- percentEncode( "/" )是按照1.b中描述的URL编码规则对字符 "/" 进行编码得到的值，即"%2F"。
- percentEncode(CanonicalizedQueryString)是对第1步中构造的规范化请求字符串按1.b中描述的URL编码规则编码后得到的字符串。

## 3. 按照RFC2104的定义，使用上面的用于签名的字符串计算签名HMAC值

。

注意：计算签名时使用的Key就是用户持有的Access Key Secret并加上一个“&”字符(ASCII:38)，使用的哈希算法是SHA1。

## 4. 按照Base64编码规则把上面的HMAC值编码成字符串，即得到签名值（Signature）。

## 5. 最后将得到的签名值作为Signature参数添加到请求参数中，即完成对请求签名的过程。



注意: 得到的签名值在作为最后的请求参数值提交给OPENSEARCH服务器的时候, 要和其他参数一样, 按照RFC3986的规则进行URL编码)

## 示例

以风险拦截为例, 签名前的请求url为:

```
https://cf.aliyuncs.com/?TimeStamp=2012-12-26T10:33:56Z&Format=XML&AccessKeyId=testid&Action=Authenticate&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2014-05-26&SignatureVersion=1.0
```

那么 StringToSign 就是:

```
GET&%2F&AccessKeyId%3Dtestid&Action%3DAuthenticate&Format%3DXML&RegionId%3Dregion1&SignatureMethod%3DHMAC-SHA1&SignatureNonce%3DNwDAxvLU6tFE0DVb&SignatureVersion%3D1.0&TimeStamp%3D2012-12-26T10%253A33%253A56Z&Version%3D2014-05-26
```

假如使用的 Access Key Id 是 “testid”, Access Key Secret 是 “testsecret”, 用于计算 HMAC 的 Key 就是 “testsecret&”, 则计算得到的签名值是:

```
SDFQNvyH5rtkc9T5Fwo8DOjw5hc=
```

签名后的请求 URL 为 ( 注意增加了 Signature 参数 ):

```
http://cf.aliyuncs.com/?TimeStamp=2012-12-26T10%3A33%3A56Z&Format=XML&AccessKeyId=testid&Action=Authenticate&SignatureMethod=HMAC-SHA1&RegionId=region1&SignatureNonce=NwDAxvLU6tFE0DVb&Version=2012-09-13&SignatureVersion=1.0&Signature=SDFQNvyH5rtkc9T5Fwo8DOjw5hc%3d
```

## SDK教程

## JAVA教程

# JAVA-SDK使用说明

## 创建Access Key

1. 登陆阿里云账号;
2. 打开 “我的Access Key” 页面，页面地址：[http://i.aliyun.com/access\\_key/](http://i.aliyun.com/access_key/);
3. 创建Access Key。

## 安装SDK

- 开发环境阿里云Java SDK支持J2SE Development Kit (JDK) 1.5或以上版本。
- 安装步骤
  - 1、在解压的文件中可以找到aliyun-java-sdk-core-2.2.5.jar和aliyun-java-sdk-afs.jar两个包；  
**说明：**您可以在云盾数据风控管理控制台>人机验证>配置管理页面，选择已添加的配置记录，单击**系统代码集成**，在服务端API集成步骤中下载对应语言的SDK包。
  - 2、以Eclipse为例导入jar包为例：在Eclipse上右键点击工程->Properties->Java Build Path->Libraries->Add External JARs ( netbeans和intellij等其他idea用户请自行按照相应导入jar包得方法导入这个包 ) ；
  - 3、选择以上jar包，点击 “OK” 按钮，经过以上步骤，你就可以在工程中使用阿里云Java SDK了。

## 快速入门

初始化一个IClientProfile

- 1、与阿里云SDK相关的类都在包com.aliyuncs.IAcsClient下；
- 2、IClientProfile是与API交互的接口，SDK的操作都是通过IClientProfile完成的；
- 3、IClientProfile可以复用，建议设置成应用程序全局唯一的；
- 4、用户可以调用带有connectTimeout和readTimeout的构造方法来设置SDK调用接口的连接超时时间和读取超时时间，SDK默认的连接超时时间是3秒，读取超时时间是80秒。

```
String regionid = "cn-hangzhou";
String accessKeyId = "**** Provide your AccessKeyId ****";
String accessKeySecret = "**** Provide your AccessKeySecret ****";

// Create a new IClientProfile instance
IClientProfile profile= DefaultProfile.getProfile(regionid, accessKeyId, accessKeySecret);
IAcsClient client = new DefaultAcsClient(profile);
DefaultProfile.addEndpoint("cn-hangzhou", "cn-hangzhou", "afs", "afs.aliyuncs.com");
```

## 示例代码

## 人机验证服务API

```
public void test(){
    AuthenticateSigRequest request = new AuthenticateSigRequest();
    request.setSessionId("xxx");// 必填参数，从前端获取，不可更改
    request.setSig("xxx");// 必填参数，从前端获取，不可更改
    request.setToken("xxx");// 必填参数，从前端获取，不可更改
    request.setScene("xxx");// 必填参数，从前端获取，不可更改
    request.setAppKey("xxx");// 必填参数，后端填写
    request.setRemotelp("xxx");// 必填参数，后端填写

    try {
        //response的代码枚举：100验签通过，900验签失败
        AuthenticateSigResponse response = client.getAcsResponse(request);
        // TODO
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## 无痕服务端代码

```
public void test(){
    AnalyzeNvcRequest request = new AnalyzeNvcRequest();
    request.setData("xxx");// 前端获取
    //注意:setScoreJsonStr方法声明"后端调用风控返回结果"与"客户端执行操作"之间的映射关系，并通知验证码服务端做二次验证授权，客户端请严格按照风控后端response+映射关系执行相应操作，否则会导致调用异常
    //例如，setScoreJsonStr声明了"400":"SC"，则后端返回400时，您客户端必须唤醒SC，若唤醒NC则会导致失败
    。
    request.setScoreJsonStr("{\"200\":\"PASS\",\"400\":\"NC\",\"600\":\"SC\",\"800\":\"BLOCK\"}");
    try {
        AnalyzeNvcResponse response = client.getAcsResponse(request);
        // TODO
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## .NET教程

## NET-SDK使用说明

## 创建Access Key

1. 登陆阿里云账号;
2. 打开“我的Access Key”页面，页面地址：[http://i.aliyun.com/access\\_key/](http://i.aliyun.com/access_key/);
3. 创建Access Key。

## 安装SDK

### - 开发环境

阿里云.Net SDK支持framework 4.0或以上版本。

### - 安装步骤

- 1、在解压的文件中可以找到aliyun-net-sdk-afs.dll和aliyun-net-sdk-core.dll这两个文件;

**说明：**您可以在云盾数据风控管理控制台>人机验证>配置管理页面，选择已添加的配置记录，单击**系统代码集成**，在服务端API集成步骤中下载对应语言的SDK包。

- 2、在你的Visual Studio项目上右键点击，添加引用->浏览（如果你用的是MonoDevelop，请自行按照相应导入dll的方法导入这个dll文件);

- 3、选择上述的dll文件，点击“确定”按钮，经过以上步骤，你就可以在工程中使用阿里云.Net SDK了。

## 快速入门

初始化一个IClientProfile

- 1、与阿里云SDK相关的类都在包com.aliyuncs.IAcsClient下；
- 2、IClientProfile是与API交互的接口，SDK的操作都是通过IClientProfile完成的；
- 3、IClientProfile可以复用，建议设置成应用程序全局唯一的；
- 4、用户可以调用带有connectTimeout和readTimeout的构造方法来设置SDK调用接口的连接超时时间和读取超时时间，SDK默认的连接超时时间是3秒，读取超时时间是80秒。

```
String regionid = "cn-hangzhou";
String accessKeyId = "**** Provide your AccessKeyId ****";
String accessKeySecret = "**** Provide your AccessKeySecret ****";

// Create a new IClientProfile instance
IClientProfile profile= DefaultProfile.getProfile(regionid, accessKeyId, accessKeySecret);
IAcsClient client = new DefaultAcsClient(profile);
DefaultProfile.addEndpoint("cn-hangzhou", "cn-hangzhou", "afs", "afs.aliyuncs.com");
```

## 示例代码

人机验证API

```
public void Test()
{
    AuthenticateSigRequest request = new AuthenticateSigRequest();
    request.SessionId = "xxx";// 必填参数，从前端获取，不可更改
    request.Sig = "xxx";// 必填参数，从前端获取，不可更改
    request.Token = "xxx";// 必填参数，从前端获取，不可更改
    request.Scene = "xxx";// 必填参数，从前端获取，不可更改
    request.AppKey = "xxx";// 必填参数，后端填写
    request.RemoteIp = "xxx";// 必填参数，后端填写

    try
    {
        //response的code枚举：100验签通过，900验签失败
        AuthenticateSigResponse response = client.GetAcsResponse(request);
        // TODO
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }
}
```

## 无痕接入代码

### 人机验证API

```
public void Test()
{
    AnalyzeNvcRequest request = new AnalyzeNvcRequest();
    request.Data = "xxx";// 必填参数，从前端获取，不可更改
    //注意:setScoreJsonStr方法声明"后端调用风控返回结果"与"客户端执行操作"之间的映射关系，并通知验证码服务端做二次验证授权，客户端请严格按照风控后端response+映射关系执行相应操作，否则会导致调用异常
    //例如，setScoreJsonStr声明了"400":"SC"，则后端返回400时，您客户端必须唤醒SC，若唤醒NC则会导致失败。
    request.ScoreJsonStr("{\"200\":\"PASS\",\"400\":\"NC\",\"600\":\"SC\",\"800\":\"BLOCK\"}");

    try
    {
        AnalyzeNvcResponse response = client.GetAcsResponse(request);
        // TODO
    }
    catch (Exception e)
    {
        Console.WriteLine(e.ToString());
    }
}
```

# PHP教程

## PHP-SDK使用说明

### 创建Access Key

1. 登陆阿里云账号;
2. 打开“我的Access Key”页面，页面地址：[http://i.aliyun.com/access\\_key/](http://i.aliyun.com/access_key/);
3. 创建Access Key。

### 安装SDK

#### - 开发环境

阿里云PHP SDK支持PHP5.3及以上版本。

#### 安装步骤

- 1、下载php\_sdk.zip，将压缩包解压后放置到系统的任意目录（需要保证有读取权限）；

**说明：**您可以在云盾数据风控管理控制台>人机验证>配置管理页面，选择已添加的配置记录，单击**系统代码集成**，在服务端API集成步骤中下载对应语言的SDK包。

- 2、在PHP文件中包含其头文件（注意文件路径要正确），并导入使用别名，如：

```
<?php
require_once 'aliyun-php-sdk-core/Config.php';
use CF\Request\V20151127 as CF;
//Do something below...
```

### 快速入门

初始化一个IClientProfile

- 1、与阿里云SDK相关的类都在包com.aliyuncs.IAcsClient下；
- 2、IClientProfile是与API交互的接口，SDK的操作都是通过IClientProfile完成的；
- 3、IClientProfile可以复用，建议设置成应用程序全局唯一的；
- 4、用户可以修改类AliyunClient里的变量 \$connectTimeout和\$readTimeout来设置SDK调用接口的连接超时时间和读取超时时间，SDK默认的连接超时时间是3秒，读取超时时间是80秒。

```
<?php
include_once './aliyun-php-sdk-core/Config.php';
use Afs\Request\V20180112 as Afs;

//YOUR ACCESS_KEY、YOUR ACCESS_SECRET请替换成您的阿里云accesskey id和secret
$iClientProfile = DefaultProfile::getProfile("cn-hangzhou", "YOUR ACCESSKEY", "YOUR ACCESS_SECRET");
$client = new DefaultAcsClient($iClientProfile);
DefaultProfile::addEndpoint("cn-hangzhou", "cn-hangzhou", "afs", "afs.aliyuncs.com");
$request = new Afs\AuthenticateSigRequest();
$request->setSessionId("xxx");// 必填参数，从前端获取，不可更改
$request->setToken("xxx");// 必填参数，从前端获取，不可更改
$request->setSig("xxx");// 必填参数，从前端获取，不可更改
$request->setScene("xxx");// 必填参数，从前端获取，不可更改
$request->setAppKey("xxx");//必填参数，后端填写
$request->setRemoteIp("xxx");//必填参数，后端填写

$response = $client->getAcsResponse($request);//response的code枚举：100验签通过，900验签失败
print_r($response);
```

## 无痕接入

### - 人机验证服务API

```
$request = new Afs\AnalyzeNvcRequest();
$request->setData("xxx");// 必填参数，从前端获取，不可更改
//注意:setScoreJsonStr方法声明"后端调用风控返回结果"与"客户端执行操作"之间的映射关系，并通知验证码服务端做二次验证授权，客户端请严格按照风控后端response+映射关系执行相应操作，否则会导致调用异常
//例如，setScoreJsonStr声明了"400":"SC"，则后端返回400时，您客户端必须唤醒SC，若唤醒NC则会导致失败。
$request->setScoreJsonStr("{\"200\":\"PASS\",\"400\":\"NC\",\"600\":\"SC\",\"800\":\"BLOCK\"}");

$response = $client->doAction($request);
print_r($response);
```

## Python教程

## Python-SDK使用说明

### 创建Access Key

1. 登陆阿里云账号;

2.打开“我的Access Key”页面，页面地址：[http://i.aliyun.com/access\\_key/](http://i.aliyun.com/access_key/);

3.创建Access Key。

## 安装SDK

- 开发环境

阿里云python支持python2.7及以上版本。

- 安装步骤

1、下载python\_sdk.zip，将压缩包解压后放置到系统的任意目录（需要保证有读取权限）。

**说明：**您可以在云盾数据风控管理控制台>人机验证>配置管理页面，选择已添加的配置记录，单击**系统代码集成**，在服务端API集成步骤中下载对应语言的SDK包。

## 快速入门

初始化一个IClientProfile

1、与阿里云SDK相关的类都在包com.aliyuncs.IAcsClient下;

2、IClientProfile是与API交互的接口，SDK的操作都是通过IClientProfile完成的;

3、IClientProfile可以复用，建议设置成应用程序全局唯一的。

```
clt = client.AcsClient(ak=accessKeyId,secret=accessKeySecret,region_id='cn-hangzhou') #阿里云颁发的appkey和secret
```

## 示例代码

- 人机验证服务API

```
request = AuthenticateSigRequest.AuthenticateSigRequest()

#必填参数：从前端获取，不可更改
request.set_SessionId('xxx')
#必填参数：从前端获取，不可更改
request.set_Sig('xxx')
#必填参数：从前端获取，不可更改
request.set_Token('xxx')
#必填参数：从前端获取，不可更改
request.set_Scene('xxx')
#必填参数：后端填写
request.set_AppKey('xxx')
#必填参数：后端填写
request.set_RemoteIp('xxx')
#response的BizCode枚举：100验签通过，900验签失败
result = clt.do_action(request)
print result
```



## 无痕接入

- 人机验证服务API

```
request = AnalyzeNvcRequest.AnalyzeNvcRequest()
```

#必填参数：从前端获取，不可更改

```
request.set_Data('xxx')
```

#注意:setScoreJsonStr方法声明"后端调用风控返回结果"与"客户端执行操作"之间的映射关系，并通知验证码服务端做二次验证授权，客户端请严格按照风控后端response+映射关系执行相应操作，否则会导致调用异常

#例如，setScoreJsonStr声明了"400":"SC"，则后端返回400时，您客户端必须唤醒SC，若唤醒NC则会导致失败。

```
request.set_ScoreJsonStr({'200':"PASS","400":"NC","600":"SC","800":"BLOCK"});
```

```
result = clt.do_action(request)
```

```
print result
```

## 控制台使用说明

## 配置管理

### 新增配置

“新增配置”可以引导您新增一条验证码相关的配置信息。您需要根据您的场景录入“服务名称”、“高峰期PV”，并选择“使用场景”、“应用类型”及“验证方式”，点击“下一步”。

**注意事项**

1.这里配置并发布后线上实时生效，不可修改，请谨慎配置。

2.线上不可存在“使用场景”、“应用类型”及“验证方式”三者完全一样的多条配置，您在配置时如果线上已有一套完全相同的配置，则会发布失败，您直接使用已有配置即可。

1 配置服务内容

2 系统代码集成&测试

3 完成

配置名称: 线上配置

使用场景: ☒ 登录 ☐ 注册 ☐ 活动 ☐ 论坛 ☐ 短信 ☐ 其它

高峰期PV: 100

应用类型: ☒ 网页 ☐ 移动端WAP/HTML5 ☐ Android ☐ IOS

① 免费用户无法使用Android、IOS应用类型, 立即前往 升级

验证方式: ☐ 滑动验证① ☐ 刮刮卡① ☐ 无感验证① ☒ 智能验证① ☐ 投票验证①

智能验证: 用户仅需一次点击, 即可进行安全验证, 更高效更安全

验证

点击开始验证

咨询 · 建议

下一步

进入“系统代码集成&测试”流程，这里提供给您前端和服务端的接入代码，请妥善保管。如果您对您页面上验证码的配置、样式等有特殊需要，也可以参照我们的文档，或使用“自定义样式”功能。

操作流程：

- 找到需要使用风险识别服务的页面，将下方示例代码复制进去，注意将form表单替换成自己的业务表单；
- 运行环境：Tomcat、Apache等；
- 接入成功校验：打开浏览器控制台（F12开发者工具），在页面上移动鼠标、点击键盘：
  - 控制台收到analyze.jsonp请求（多个）；
  - 在Headers - Query String Parameters 里看到：n、a、t、asyn、scene等参数；
  - 双击analyze.jsonp请求，获得onJSONPcallback({"result":0,"success":true})。

使用场景-登录：

复制代码

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<script>
window.NVC_Opt = {
  appkey: '9P9T',
  scene: 'ic_activity_h5',
  renderTo: '#captcha',
  trans: {'key1': 'code0', 'nvcCode': '200'},
  elements: [
    '//img.alicdn.com/tfs/TB17cwlsL8KJly0FnXXcFDpXa-50-74.png',
    '//img.alicdn.com/tfs/TB17cwlsL8KJly0FnXXcFDpXa-50-74.png'
  ],
  bg_back_prepared: '//img.alicdn.com/tfs/TB1skE5SFXXXb3XXXXXXX-100-80.png',
  bg_front:
'data:image/png;base64,iVBORw0KGgoAAAANSUUhEugAAAGQAAABQCAAAADY1yDdAAAABGdBtUEALGPC/xhBQAAAFzUkdCAK7OHOKAAAAADUEXURefkSw+ruswAAA
ATSURBVFD7cExAQAAAMKg9U9tCU+gAAAAAAAIc3AR+QAAFPiUGoAAAAAEIFTksQmCC',
  obj_ok: '//img.alicdn.com/tfs/TB1rmyTtF8KJly0FeXXXKEXXa-50-74.png',
  bg_back_pass: '//img.alicdn.com/tfs/TB1KDxCsvXXXXXasXFXXXXXXX-100-80.png',
  obj_error: '//img.alicdn.com/tfs/TB1q9yTtF8KJly0FeXXXKEXXa-50-74.png',
  bg_back_fail: '//img.alicdn.com/tfs/TB1w2oSFXXXb4XpXXXXXXX-100-80.png',
```

可以在此处获取到各版本的SDK DEMO

1 服务端API集成

操作流程：

1. 下载对应语言的SDK: [Python版本](#) [.Net版本](#) [JAVA版本](#) [PHP版本](#)

2. 将SDK加载到工程中；

3. 参考下方的代码，开发第一步页面请求的处理类（如java的ActionmController,Servlet）；注意填入自己的阿里云accesskey和secret；

4. 参考下方代码，对运行结果进行处理；请注意对服务端出现的错误进行兼容处理。

调用代码示例：

JAVA

Python

.Net

PHP

```
package com.alibaba.test;

import com.aliyuncs.DefaultAcsClient;
import com.aliyuncs.IAcsClient;
import com.aliyuncs.afs.model.v20180112.AuthenticateSigRequest;
import com.aliyuncs.afs.model.v20180112.AuthenticateSigResponse;
import com.aliyuncs.profile.DefaultProfile;
import com.aliyuncs.profile.IClientProfile;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
```

复制代码

点击“提交”之后即可完成。

1 配置服务内容

2 系统代码集成&测试

3 完成

✔

恭喜您，配置已经成功创建

可在“配置管理”中查看已创建的配置，或进行[样式自定义](#)

返回

## 配置管理

“配置管理”页面可以看到您当前所有发布的配置选项，包括配置名称、appkey、验证方式、应用类型、使用场景等。

配置管理

数据监控

预警设置

新增配置

配置名称	appkey	验证方式	应用类型	使用场景	最后更新	操作
活动_滑动验证_H5	9P9T	滑动验证	H5	活动	2018-02-01 13:48:00	<a href="#">自定义样式</a>
登录_滑动验证_PC	9P9T	滑动验证	PC	登录	2018-02-01 13:48:00	<a href="#">自定义样式</a>
注册_滑动验证_PC	9P9T	滑动验证	PC	注册	2018-02-01 13:48:00	<a href="#">自定义样式</a>
活动_滑动验证_PC	9P9T	滑动验证	PC	活动	2018-02-01 13:48:00	<a href="#">自定义样式</a>
论坛_滑动验证_H5	9P9T	滑动验证	H5	论坛	2018-02-01 13:48:01	<a href="#">自定义样式</a>
论坛_滑动验证_PC	9P9T	滑动验证	PC	论坛	2018-02-01 13:48:01	<a href="#">自定义样式</a>
登录_滑动验证_H5	9P9T	滑动验证	H5	登录	2018-02-01 13:48:00	<a href="#">自定义样式</a>
注册_滑动验证_H5	9P9T	滑动验证	H5	注册	2018-02-01 13:48:00	<a href="#">自定义样式</a>

## 自定义样式

“自定义样式”功能可以引导您对各种验证码进行个性化的配置，您可以按照页面的引导自定义颜色、大小、图标、文案等各式内容。配置页面右边会在您每次修改后实时渲染配置效果。当年您觉得满意后，请点击“生成代码”按钮，并将下方为您自定义生成的代码妥善保存并对照接入。

### 注意事项

- 自定义配置项中的图片资源需要您自己维护可用的web访问。
- 针对刮刮卡交互验证码配置项中，“正常状态前景图片”（即浮层）必须为base64的图片数据流，不接受外部资源。
- 针对“自定义样式”功能无法满足的特殊配置，您也可以自行参照用户文档中对应的前端代码集成文档中的“自样式自定义”部分进行参数、样式的修改。

配置名称: 测试配置1      APPKey: YJAM      验证方式: 刮刮卡

样式配置:

浮层大小: 宽 300 高 125

正常状态前景图片: data:image/png;base64,VBOR9

正常状态背景图片: //img.alicdn.com/tps/TB1skE5S

验证失败状态背景图片: //img.alicdn.com/tps/TB1w2oOI

验证成功状态背景图片: //img.alicdn.com/tps/TB1KDXCI

效果预览: 验证通过状态

请在屏幕上滑动, 刮出两面盾牌

生成代码

样式代码:  
请妥善保管配置好的样式代码, 此处未提供样式代码的保存功能

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<meta name="description" content="">
<meta name="keywords" content="">
<meta name="apple-mobile-web-app-capable" content="yes">
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta name="format-detection" content="telephone= tdi:{vertical-align:top;}no">
<meta name="format-detection" content="email=no">
<meta name="data-spm" content="a2145">
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=0">

<title>test-H5</title>
<!-- <script charset="utf-8" src="/js/ch5.js"></script> -->
<script charset="utf-8" src="//g.alicdn.com/sd/nch5/index.js"></script>
</head>
<body>
<div id="gc"></div>
```

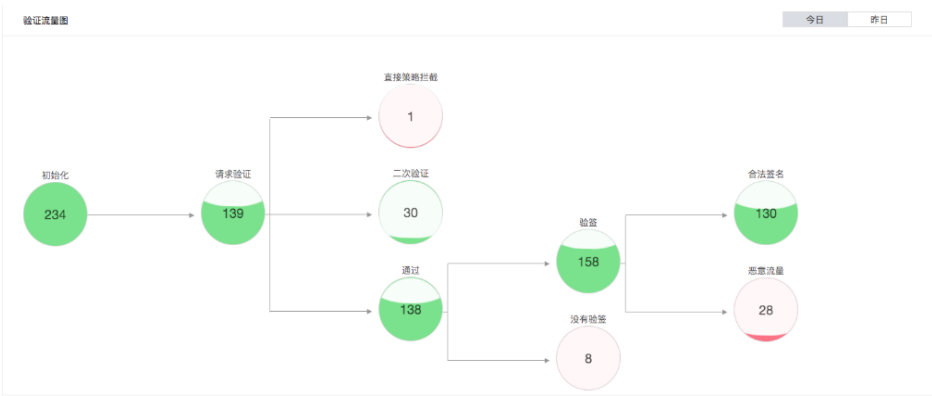
# 数据监控

## 验证数据

### 1.验证流量图

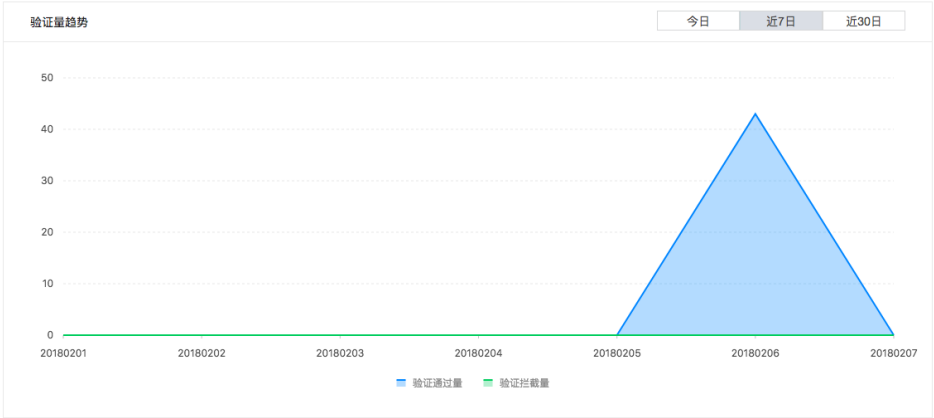
“验证流量图” 为您展示您当前选择的配置名称对应的验证码整条链路每个节点的数据流分布情况，您可以在 此直观的看到当前场景下验证码的攻防情况，并可以点击右上角的“昨日/今日” 切换显示。

注意：根据线上攻击情况，此处有可能出现下游数据量级大于上游的情况（如，您的业务接口直接遭受了接口扫描，攻击者请求中验证码sig签名串字段全部为空，则“验签” 量可能会大于“通过” 量）



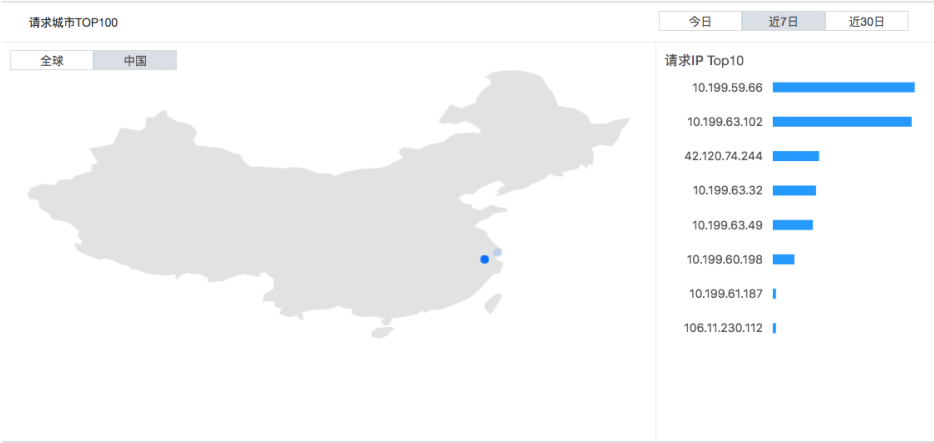
### 2.验签量趋势、验签量趋势

“验证量趋势”、“验签量趋势” 为您展示您当天/7天/30天，验证码接口和验签接口的通过量/拦截量数据波动情况，您可以在这里直观的看到当前场景下验证码历史攻防情况。



3.请求城市与请求IP

“请求城市与请求IP” 为您展示的是验证码接口在国内、全球范围内请求来源的情况与Top10的请求IP，您可以综合看到今天/7天/30天的请求分布。



攻击分析

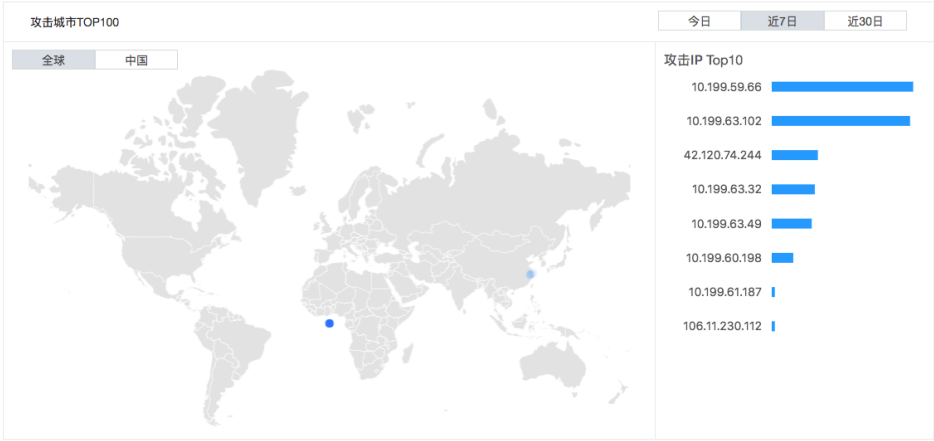
1.风险等级与攻击次数

验证码服务端报表系统会通过算法综合统计您本月、上月遭受攻击的次数，并据此推断出您线上人机攻防压力情况，帮助您进行综合的业务决策。



2.攻击城市与攻击IP

“攻击城市与攻击IP” 为您展示的是验证码接口拦截的恶意请求在国内、全球范围内请求来源的情况与Top10的攻击IP，您可以综合看到今天/7天/30天的攻击分布。



## 预警设置

此处为您归纳了线上比较通用的几种预警逻辑，您选择性的开启。每种预警如果触发，每天会根据发送时间推送一次到您配置的发送渠道。

拦截量暴涨

开启

预警内容： 同比昨天增加300%且15分钟拦截量超过500

预警频率： 每天一次

预警发送渠道：☐ 站内信 ☐ 短信 ☐ 邮件

预警发送时间：☒ 全天任意时间 ☐ 自定义时间段

请求量暴涨

开启

预警内容： 同比昨天增加300%且15分钟拦截量超过500

预警频率： 每天一次

预警发送渠道：☐ 站内信 ☐ 短信 ☐ 邮件

预警发送时间：☒ 全天任意时间 ☐ 自定义时间段

爆破验证码接口

开启

预警内容： analyze尝试量>初始化量 && analyze尝试量>800

预警频率： 每天一次

预警发送渠道：☐ 站内信 ☐ 短信 ☐ 邮件

预警发送时间：☒ 全天任意时间 ☐ 自定义时间段

爆破业务接口

开启

预警内容： sig尝试量>验证码通过量 && sig\_block>300

预警频率： 每天一次

预警发送渠道：☐ 站内信 ☐ 短信 ☐ 邮件

预警发送时间：☒ 全天任意时间 ☐ 自定义时间段

## 错误码

### 错误码

#### 客户端错误码

错误码	值	含义
-----	---	----

SEC_ERROR_SECURITYBODY_INVALID_PARAM	1401	参数不正确，请检查输入的参数
SEC_ERROR_SECURITYBODY_ENCRYPTION_ERROR	1407	数据加密错误，请确保使用正确的图片yw_1222_0335.jpg，请联系我们
SEC_ERROR_SECURITYBODY_DATA_FILE_MISMATCH	1411	图片yw_1222_0335.jpg与sdk之间不匹配，请确保使用的sdk和图片来自同一个下载的zip
SEC_ERROR_SECURITYBODY_NO_DATA_FILE	1412	图片yw_1222_0335.jpg在App中未找到，请检查是否正确引入图片，另外如果设置gradle关于shrinkResource相关配置后，请keep住图片yw_1222_0335.jpg否则打包时图片yw_1222_0335.jpg会被优化，详情请见： <a href="http://tools.android.com/tech-docs/new-build-system/resource-shrinking">http://tools.android.com/tech-docs/new-build-system/resource-shrinking</a>
SEC_ERROR_SECURITYBODY_INCORRECT_DATA_FILE	1413	非法的图片yw_1222_0335.jpg请检查该图片正确性，确保和下载的sdk的zip中是一致的。
SEC_ERROR_SECURITYBODY_KEY_NOT_EXSITED	1414	图片yw_1222_0335.jpg中缺少相关数据，请重新下载图片。
SEC_ERROR_SECURITYBODY_LOW_VERSION_DATA	1415	非法的图片yw_1222_0335.jpg图片是低版本的，请下载新的图片和sdk相匹配
SEC_ERROR_SECURITYBODY_UNKNOWN	1499	未知错误，请联系我们
SEC_ERROR_INIT_PUBLICKEY_FIND_ERROR	105	RSA文件不正常，无法正常解析出公钥
SEC_ERROR_INIT_SO_NOT_EXIST	106	在APK中找不到对应aebi的libsecurityguard***.so文件，请确认存在此文件，并且它的版本和jar包的版本号一致。（不同版本的不通用，请不要拿其他版本的替换）
SEC_ERROR_INIT_DECODESO_FAIL	107	libsecurityguard***.so解码失败，请卸载后全新安装或检查磁盘空间
SEC_ERROR_INIT_LOADSOINNER_FAILED	108	加载内部so失败
SEC_ERROR_INIT_FDSOFUN_FAILED	109	内部so运行失败
SEC_ERROR_INIT_PLUGIN_NOT_EXISTED	110	插件不存在，请检查打包配置中，so是否正确打入APK中
SEC_ERROR_INIT_PLUGIN_LOAD_FAILED	111	加载插件失败，一般不会发生。请检查是否存在IO异常或内存

		分配不足
SEC_ERROR_INIT_LOAD_INTERFACE_NOT_EXISTED	112	获取接口失败，请检查插件so版本是否与JAR包版本是否匹配
SEC_ERROR_PLUGIN_REQUIREMENT_NOT_MEET	113	插件依赖不匹配，请检查插件版本，查看依赖关系是否兼容
SEC_ERROR_INIT_EXTRACT_DIR_NOT_EXISTED	114	系统IO异常，插件加载目录打开失败
SEC_ERROR_INIT_DATA_FILE_MISMATCH	121	图片文件有问题。一般是生成图片文件时WSG上注册的应用公钥和当前应用的公钥信息不一致。
SEC_ERROR_INIT_NO_DATA_FILE	122	没有找到图片文件，请确保图片文件在项目目录下
SEC_ERROR_INIT_INCORRECT_DATA_FILE	123	图片文件格式有问题，请重新生成图片文件。一种常见场景就是二方和三方图片混用。二方和三方的图片不兼容，需要各自生成。还有一个场景是4.x.x的SDK使用了5的图片，v5图片只能在6.x.x上使用
SEC_ERROR_INIT_LOW_VERSION_DATA	124	当前图片的版本太低
SEC_ERROR_INIT_PARSE_USER_CONFIG_ERROR	125	init with authcode 初始化错误，请联系答疑账户具体定位
SEC_ERROR_INIT_UNKNOWN_ERROR	199	未知错误，请重试

## 网关错误码

错误码	值	含义
InvalidTimeStamp.Expired	400	时间戳超时，建议检查服务器时间
SignatureNonceUsed	400	重复请求
Invalid{ParameterName}	400	指定参数不合法
InternalError	500	服务内部错误，建议联系客户人员
Throttling.User	400	超过流量限制，用户被限流
Throttling.Api	400	超过流量限制，用户调用的api被限流
Throttling.Overflow	400	资源包已耗尽，请购买
MissingSecurityToken	400	缺少AccessKeyId，建议检查代码中的ACCESS_KEY、



		ACCESS_SECRET是否合法
Speicified endpoint or uri is not valid	ClientException	请检查网络访问 jaq.aliyuncs.com是否畅通
Specified access key is not found	ClientException	请查看ACCESS_KEY、 ACCESS_SECRET是否正确

数据风控错误码

错误码	值	含义
[ParameterName] is null	400	必填参数为空，建议补全相关参数。
NOT_AUTHORITY with sdkToken	400	sdk版只有相应付费用户才可以使用，请购买
Resource package fully used	400	资源包耗尽，请购买
Invalid parameter	400	参数不合法
The uid do not create the service	400	服务未创建，请先到数据风控控制台发布服务
The uid do not apply the service	400	服务未创建，请先到数据风控控制台发布服务
null jsToken and null sdkToken	400	jstoken或sdktoken必带其一
jstoken is invalid	400	jstoken不合法，可能为攻击请求
sdktoken is invalid	400	sdktoken不合法，可能为攻击请求
user not event	500	请到数据风控控制台发布对应事件

人机验证错误码

错误码	值	含义
invalid sig parameter	400	非法签名串，一般为攻击者请求
[ParameterName] is invalid	400	签名串格式不合法，一般为攻击者伪造请求或接入代码有问题
Invalid parameter	400	签名串格式不合法，一般为攻击者伪造请求或接入代码有问题
Missing{ParamterName} : {ParamterName} is mandatory for this action.	ClientException	缺少必要参数，请检查接入代码
currentUrl format is error	400	请求的参数url合法性检查失败导致