ALIBABA CLOUD

阿里云

数据风控

无痕验证

文档版本：20201230

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.前端代码集成（新版）

1.1. 无痕验证集成方式

无痕验证是阿里巴巴集团提供的全新的人机验证解决方案。由无痕验证组件收集并统一调度下游验证码服务实现综合人机对抗的识别与处理。

在业务中接入无痕验证后，组件完全静默采集所需信息，从而对大部分正常用户完全“零打扰”，仅对存在风险的用户根据风险等级返回相应的下游验证码服务的唤醒建议（即调用该风险等级相应的验证码服务进行人机验证）。

前端接入代码示例

在需要使用无痕验证功能的Web前端页面中，首先需要对无痕验证进行初始化，然后需要主动获取人机信息串并传给业务服务端，业务服务端通过风险服务端进行查询后，返回结果给予客户端，客户端根据风险等级判断无痕验证通过，拦截，及是否需要唤醒二次验证。以下代码为无痕验证功能的前端接入代码示例。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <script src="https://g.alicdn.com/AWSC/AWSC/awsc.js"></script>
</head>
<body>
<form id="login-form">
  <input id="account" type="text" placeholder="账号" />
  <input id="password" type="password" placeholder="密码">
  <button type="button" id="register" >注册</button>
</form>
<div id="nc"></div>
<script>
  var btn = document.getElementById("register");
  // 实例化nvc 对无痕验证进行初始化操作
  AWSC.use("nvc", function (state, module) {
    // 初始化 调用module.init进行初始化
    window.nvc = module.init({
      // 应用类型标识。它和使用场景标识（scene字段）一起决定了无痕验证的业务场景与后端对应使用的策略模型。您可以在人机验证控制台的配置管理页签找到对应的appkey字段值，请务必正确填写。
      appkey: "CF_APP_1",
      //使用场景标识。它和应用类型标识（appkey字段）一起决定了无痕验证的业务场景与后端对应使用的策略模型。您可以在人机验证控制台的配置管理页签找到对应的scene值，请务必正确填写。
      scene: "nvc_register",
      // 二次验证获取人机信息串，跟随业务请求一起上传至业务服务器，由业务服务器进行验签。
      success: function (data) {
```

```
          window.console && console.log(data)
        },
        // 前端二次验证失败时触发该回调参数
        fail: function (failCode) {
          window.console && console.log(failCode)
        },
        // 前端二次验证加载异常时触发该回调参数。
        error: function (errorCode) {
          window.console && console.log(errorCode)
        }
      });
      // 绑定事件
      btn.onclick = onclick;
    });
    // 发送业务请求：点击按钮时触发，主动获取人机信息串，并发送给业务服务端
    function onclick() {
      window.nvc.getNVCValAsync(function (nvcVal) {
        var s = document.createElement('script');
        // 获取人机信息串
        // 将以下getNVCVal()函数的值，跟随业务请求一起上传，由后端请求AnalyzeNvc接口并返回200，400，600或者800。
        // 正式上线前务必将该服务端接口，更改为您自己的业务服务端接口
        s.src = 'https://cf.aliyun.com/nvc/nvcAnalyze.jsonp?callback=yourRegisterRequest&a=' + nvcVal;
        document.body.append(s);
      });
    }
    // 处理业务返回结果：人机信息串上传接口的回调函数，通过业务服务端的返回结果，控制无痕验证的不同状态。
    function yourRegisterRequest(json) {
      // 业务服务器请求回调控制是否需要二次验证
      if (json.result.code === 100 || json.result.code === 200) {
        // 无痕验证通过
        alert("register success!")
      } else if (json.result.code === 800 || json.result.code === 900) {
        // 无痕验证失败，直接拦截
        alert("register failed!")
      } else if (json.result.code === 400) {
        // 无痕验证失败，触发二次验证
        // 二次验证码（滑动验证码）配置项设置，详情请见滑动验证集成方式文档
        // 二次验证的appkey，scene，test值及success，fail，error的回调由nvc初始化时决定，请不要在二次验证时传入
        var ncoption = {
```

```
        // 声明滑动验证需要渲染的目标ID。
        renderTo: "nc",
      }
      // 唤醒二次验证（滑动验证码）
      window.nvc.getNC(ncoption);
    }
  }
</script>
</body>
</html>
```

初始化

- **资源引入**

  在Web页面中使用无痕验证功能，需要在前端页面代码中添加以下代码，引入所需的JS资源。`<script src="https://g.alicdn.com/AWSC/AWSC/awsc.js"></script>`。

- **初始化代码说明**

```
AWSC.use("nvc", function (state, module) {
    var nvcOption = {} // nvc初始化参数对象
    // 初始化 调用module.init进行初始化
    window.nvc = module.init(nvcOption);
  });
```

> **说明**
>
> 初始化`module.init`完成后，会返回一个实例化的nvc对象。示例代码中该实例化的nvc对象存在`window.nvc`变量中。

- **初始化参数说明**

  在前端代码实例化验证组件时提供了以下初始化参数，您可以根据业务需要在代码中调整这些参数。

参数	含义	是否必填
appkey	应用类型标识。它和使用场景标识（scene字段）一起决定了无痕验证的业务场景与后端对应使用的策略模型。您可以在人机验证控制台的配置管理页签找到对应的appkey字段值，请务必正确填写。	是

scene	使用场景标识。它和应用类型标识（appkey字段）一起决定了无痕验证的业务场景与后端对应使用的策略模型。您可以在人机验证控制台的配置管理页签找到对应的scene值，请务必正确填写。	是
test	测试字段，用于测试验证码的不同状态。	否
success	二次验证成功后获取人机信息串	否
fail	二次验证失败时触发该回调参数	否
error	二次验证出现错误时触发该回调参数	否

发起业务请求

在无痕验证初始化完成后，需要您主动调用nvc对象的**getNVCVal**方法获取人机信息串，并将其随业务请求一起传递至您的业务服务端，并参见服务端代码集成文档中的说明发起“业务服务端->风控服务端”的风险查询请求。

```
function onclick() {
    // 触发绑定事件后，主动调用getNVCVal方法
    window.nvc.getNVCValAsync(function (nvcVal) {
      var s = document.createElement('script');
      // 将以下getNVCVal()函数的值，跟随业务请求一起上传，由后端请求AnalyzeNvc接口并返回200，400，600或者800。
      // 正式上线前务必将该服务端接口，更改为您自己的业务服务端接口
      s.src = 'https://cf.aliyun.com/nvc/nvcAnalyze.jsonp?callback=yourRegisterRequest&a=' + nvcVal;
      document.body.append(s);
    });
  }
```

> **说明**
>
> 务必将上述示例中维护的默认接口 `https://cf.aliyun.com/nvc/nvcAnalyze.jsonp` 替换为您自己的业务请求接口（默认接口为线上Demo测试接口，不具备任何实际攻防能力）。

处理业务返回结果

在获取到风险查询返回结果后，您可以直接返回给客户端，客户端根据风险等级选择放行、拦截或调用nvc对象的getNC方法在指定div上唤醒二次验证。如果您的业务服务端自身具有风控策略（例如，黑、白名单），您也可以将您自身业务中的风控策略与人机验证服务接口二者得到的结果进行综合，最终决定客户端的风险处置方式。

```
function yourRegisterRequest(json) {
    // 根据业务服务器返回，控制无痕验证放行，拦截，触发二次验证
    if (json.result.code === 100 || json.result.code === 200) {
      // 无痕验证通过
      alert("register success!")
    } else if (json.result.code === 800 || json.result.code === 900) {
      // 无痕验证失败，直接拦截
      alert("register failed!")
    } else if (json.result.code === 400) {
      // 无痕验证失败，触发二次验证
      var ncoption = {} // ncoption为二次验证初始化参数对象，详情请参考滑动验证初始化参数
      // 唤醒二次验证（滑动验证码）
      window.nvc.getNC(ncoption);
    }
  }
```

> **说明**
>
> 二次验证的初始化参数详情参考滑动验证码初始化参数列表，且二次验证初始化参数中的appkey，scene，test值及success，fail，error等回调由nvc初始化时决定，请不要在二次验证中传入。

测试

在将前端接入代码集成至Web页面后，建议您在正式上线前通过以下方法进行测试。

1. **通过设定初始化参数test的值复现无痕验证的各个状态工作机制**

   您可以通过将test字段值设置为无痕验证服务提供的不同内容，来测无痕试验证各个状态的效果。通过配置前端页面初始化代码中的test字段来直接控制无痕验证服务接口的返回结果（通过、拦截）。例如，当您使用代码示例中提供的 `http://cf.aliyun.com/nvc/nvcAnalyze.jsonp` 服务端接口进行测试时，test字段值与无痕验证服务接口返回的结果关系如下表所示：

test字段值	模拟效果
module.TEST_PASS	无痕验证通过
module.TEST_BLOCK	无痕验证未通过，直接拦截

module.TEST_NC_PASS	唤醒滑动验证，且滑动验证通过
module.TEST_NC_BLOCK	唤醒滑动验证，且滑动验证不通过

通过这样的方式，您可以直观地观察验证码各个状态的交互、样式和流程。

> 说明
>
> - 前缀 `module` 为 `AWSC.use` 返回的对象，示例代码中命名为 `module` 。
> - 在正式上线前务必删除集成代码中的test配置项，避免出现不必要的安全问题。新版集成代码中token是自动生成的，无需用户自行指定token。如果您在正式上线或进行完整的功能性测试时未提前删除test配置项，会导致token异常，从而可能导致集成操作中某个环节出现异常，例如：验签失败等。

测试代码示例：

```
AWSC.use("nvc", function (state, module) {
  window.nvc = module.init({
    ...
    ...
    // 该配置项为测试项 在仅用来测试验证码不同状态时使用。上线时请将其删除. 无痕验证test配置项有4种不同的值对应不同的验证码状态，具体请参考文中参数定义说明部分。
    test: module.TEST_PASS, // 测试无痕验证通过
    // test: module.NVC.TEST_BLOCK, // 测试无痕验证失败
    // test: module.TEST_NC_PASS, // 唤醒二次验证（滑动验证），且二次验证通过
    // test: module.TEST_NC_BLOCK, // 唤醒二次验证（滑动验证），且二次验证失败
    ...
    ...
  });
});
```

2. 进行功能性测试和兼容性测试。

   在正式上线前，建议您执行完整的功能性测试和兼容性测试。在测试过程中，请注意：

   - 务必在前端代码中将代码示例提供的 `http://cf.aliyun.com/nvc/nvcAnalyze.jsonp` 服务端接口替换为您自己的业务接口，并按照无痕验证服务端代码集成在服务端接入人机验证服务。
   - 确保测试过程中，使用云盾人机验证控制台中为您分配的appkey和scene值进行测试。
   - 对于Internet Explorer浏览器，验证组件最低支持至Internet Explorer 9，Internet Explorer 8及以下不支持。
   - 新版集成代码中token是自动生成的，无需用户自行指定token。如果您在正式上线或进行完整的功能性测试时未提前删除test配置项，会导致token异常，从而可能导致集成操作中某个环节出现异常，例如：验签失败等。

1.2. 自定义文案

您可以自定义页面中由无痕验证唤醒的二次验证组件的显示文案。

自定义二次验证文案

您可以通过初始化参数中的**upLang**参数来自定义二次验证组件的文案。

```
AWSC.use("nvc",function(state,module){
 window.nvc = module.init({
  ...
 // 二次验证文案配置
  upLang: {
   'cn': {
     //加载状态提示。
     'LOADING': "加载中...",
     //等待滑动状态提示。
     'SLIDE': "请向右滑动验证",
     //验证通过状态提示。
     'SUCCESS': "验证通过",
     //验证失败触发拦截状态提示。
     'ERROR': "非常抱歉，网络出错了...",
     //验证失败触发拦截状态提示。
     'FAIL': "验证失败，请重试"
    }
  }
  ...
 })
})
```