

滑块验证码

概述

验证方式

本验证码接口支持**滑动拼图**、**文字点选**和**图形点选**三种验证方式：

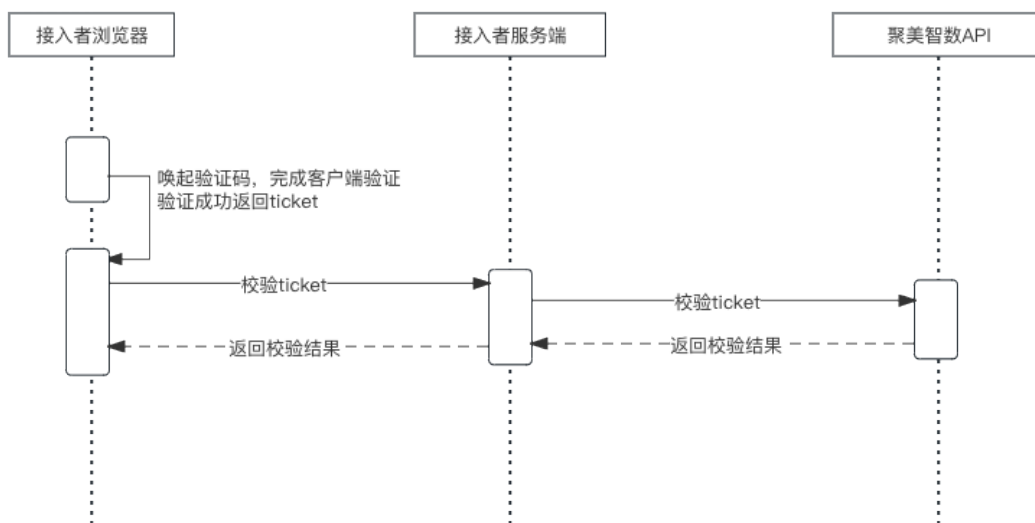
1. **滑动拼图**：轻轻一滑即刻完成验证，适用于追求用户体验的业务场景
2. **文字点选**：顺序点击图中文字即可完成验证，提高机器识别难度的同时，保证真实用户可读
3. **图形点选**：依次点击图中图形，对抗空间大，安全系数极高

开通本服务时，需要向服务商告知使用哪种验证方式

业务流程

本验证码接入包括**客户端（浏览器）接入**和**服务端接入**

1. 客户端完成滑块动作，滑块正确则返回 ticket；
2. 客户端完成验证之后，再用客户端返回的 ticket，调用服务端校验 ticket 接口完成验证。



演示链接: <https://api-js.jumdata.com/action-captcha/demo>

客户端接入

准备工作

Web 页面需引入验证码的 js, 在业务需要验证时, 唤起验证码进行验证。

```
<script type="text/javascript" src="https://api-js.jumdata.com/action-captcha/action-captcha.min.js"></script>
```

必须动态引入验证码 JS。如使用本地缓存, 或通过其他手段规避动态加载, 会导致验证码无法正常更新, 对抗能力无法保证, 甚至引起误拦截。

示例代码

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>验证码示例 - 聚美智数</title>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- 验证码程序依赖(必须)。请勿修改以下程序依赖，如使用本地缓存，或通过其他手段规避加载，
    会导致验证码无法正常更新，对抗能力无法保证，甚至引起误拦截。 -->
    <script type="text/javascript" src="https://api-js.jumdata.com/action-
    captcha/action-captcha.min.js"></script>
  </head>

  <body>
    <button id="btnCaptcha" type="button">验证</button>
  </body>

  <script type="text/javascript">
    /**
     * 回调函数，在滑块停止时触发
     */
    function callback(captchaRes) {
      // captchaRes 属性如下：
      // ret      Int      验证结果，0：验证成功。2：用户主动关闭验证码。
      // ticket   String   验证成功的票据，当且仅当 ret = 0 时 ticket 有值。
      // bizState Any     自定义透传参数。
      // randstr  String   本次验证的随机串，后续票据校验时需传递该参数。
      console.log('callback:', captchaRes)

      if(captchaRes.ret === 0){
        console.log('客户端验证成功')
        // todo 此处需要调用接入者的服务端接口，完成验证，详见服务端接入说明
      } else {
        console.log('验证失败')
      }
    }
  }

  var captcha = new JumeiActionCaptcha('您的验证码 CaptchaAppId（联系服务商开通）',
  callback, {})

  // 定义验证码触发事件
  document.getElementById('btnCaptcha').onclick = function () {
    captcha.show()
  }

```

```
    }  
  </script>  
</html>
```

如采用 Vue 开发，则可以动态引入验证码的 js

```
<template>  
  <div>  
    <button type="button" @click="open">验证</button>  
  </div>  
</template>  
<script>  
const CAPTCHA_APP_ID = '您的验证码 CaptchaAppId（联系服务商开通）';  
const SLIDER_CAPTCHA_JS = 'https://api-js.jumdata.com/action-captcha/action-captcha.min.js'  
  
export default {  
  data() {  
    return {  
      captcha: null  
    }  
  },  
  created() {  
  },  
  mounted() {  
    // 动态引入验证码的 js  
    let script = document.createElement('script');  
    script.type = 'text/javascript';  
    script.src = location.protocol + '//' + location.host + rootPath +  
SLIDER_CAPTCHA_JS;  
    let head = document.head || document.getElementsByTagName('head')[0];  
    head.appendChild(script)  
  },  
  methods: {  
    open() {  
      if (!this.captcha) {  
        // 因为 mounted 的时候验证码的 js 还没加载，因此 JumeiActionCaptcha 实例必须在 open  
的时候创建  
        this.captcha = new JumeiActionCaptcha(CAPTCHA_APP_ID, this.callback, {})  
      }  
      this.captcha.show()  
    }  
  }  
}
```

```

},
callback(captchaRes) {
  // 属性如下:
  // ret      Int      验证结果, 0: 验证成功。2: 用户主动关闭验证码。
  // ticket   String   验证成功的票据, 当且仅当 ret = 0 时 ticket 有值。
  // bizState Any      自定义透传参数。
  // randstr  String   本次验证的随机串, 后续票据校验时需传递该参数。
  console.log('callback:', captchaRes)

  if(captchaRes.ret === 0){
    console.log('客户端验证成功')
    // todo 此处需要调用接入者的服务端接口, 完成验证, 详见服务端接入说明
  } else {
    console.log('验证失败')
  }
}
}
}
</script>

```

JumeiActionCaptcha 使用说明

参数说明

参数名	值类型	说明
captchaAppId	String	验证码 CaptchaAppId (联系服务商开通)。注意: 不可使用客户端类型为小程序的 captchaAppId, 会导致数据统计错误。
callback	Function	验证码回调函数, 详见 callback 回调函数。
options	Object	验证码外观配置参数, 详见 options 外观配置参数。

callback 回调函数

验证结束后，会调用业务传入的回调函数，并在第一个参数中传入回调结果。回调结果字

段说明如下：

回调函数字段说明

字段名	值类型	说明
ret	Int	验证结果，0：验证成功。2：用户主动关闭验证码。
ticket	String	验证成功的票据，当且仅当 ret = 0 时 ticket 有值。
bizState	Any	自定义透传参数。
randstr	String	本次验证的随机串，后续票据校验时需传递该参数。
errorCode	String	错误 code ，详情请参见 回调函数 errorCode 说明。
errorMessage	String	错误信息。

回调函数 errorCode 说明

errorCode	说明
1001	加载错误

errorCode	说明
1002	调用 show 方法超时
1003	中间 js 加载超时
1004	中间 js 加载错误
1005	中间 js 运行错误
1006	拉取验证码配置错误/超时
1007	iframe 加载超时
1008	iframe 加载错误
1009	jquery 加载错误
1010	滑块 js 加载错误
1011	滑块 js 运行错误
1012	刷新连续错误 3 次
1013	验证网络连续错误 3 次

options 外观配置参数

options 参数用于对验证码进行定制外观设置，默认可以设置为空。

1、验证码弹窗内部不支持调整样式大小，如果需要调整，可在弹窗最外层用

`class=tcaptcha-transform` 的元素设置 `transform:scale()`。验证码更新可能会改变元素的 `id`, `class` 等属性，请勿依赖其他验证码元素属性值覆盖样式。

2、如果手机原生端有设置左右滑动手势，需在调用验证码 `show` 方法前禁用，验证完成后打开，防止与验证码滑动事件冲突。

配置名	值类型	说明
<code>bizState</code>	Any	自定义透传参数，业务可用该字段传递少量数据，该字段的内容会被带入 <code>callback</code> 回调的对象中。
<code>enableDarkMode</code>	Boolean/String	开启自适应深夜模式： { "enableDarkMode": true } 强制深夜模式：{ "enableDarkMode": 'force' }
<code>sdkOpts</code>	Object	示例 { "width": 140, "height": 140 } 仅支持移动端原生 <code>webview</code> 调用时传入，用来设置验证码 <code>loading</code> 加载弹窗的大小（注意，并非验证码弹窗大小）。
<code>ready</code>	Function	验证码加载完成的回调，回调参数为验证码实际的宽高： { "sdkView": { "width": number, "height": number } } 该参数仅为查看验证码宽高使用，请勿使用此参数直接设定宽高。
<code>needFeedBack</code>	Boolean/String	隐藏帮助按钮或自定义帮助按钮链接。隐藏帮助按钮： { "needFeedBack": false } 自定义帮助链接：{ "needFeedBack": 'url 地址' }

配置名	值类型	说明
userLanguage	String	指定验证码提示文案的语言，优先级高于控制台配置。支持传入值同 navigator.language 用户首选语言，大小写不敏感。详情参见 userLanguage 配置参数。
type	String	定义验证码展示方式。popup（默认）弹出式，以浮层形式居中弹出展示验证码。embed 嵌入式，以嵌入指定容器元素中的方式展示验证码。

userLanguage 配置参数

参数名	说明
zh-cn	简体中文
zh-hk	繁体中文（中国香港）
zh-tw	繁体中文（中国台湾）
en	英文
ar	阿拉伯语
my	缅甸语
fr	法语
de	德语

参数名	说明
he	希伯来语
hi	印地语
id	印尼语
it	意大利语
ja	日语
ko	朝鲜语
lo	老挝语
ms	马来语
pl	波兰语
pt	葡萄牙语
ru	俄语
es	西班牙语
th	泰语
tr	土耳其语
vi	越南语

服务端接入

客户端验证成功之后，需要由接入者的服务端再调用服务商的服务端接口完成校验，本接口即完成该功能

请求地址

以实际云市场上提供请求 `url` 为准

请求方式

- POST

请求格式

- application/x-www-form-urlencoded

请求参数

名称	类型	必须	说明
captchaAppId	String	是	验证码 captchaAppId ，和客户端一致。联系服务商申请获取
ticket	String	是	验证码返回给用户的票据

名称	类型	必须	说明
randstr	String	是	验证票据需要的随机字符串
captchaSign	String	是	验证码签名，详见验证码签名说明
ip	String	否	用户操作来源的外网 IP

验证码签名说明

```
sign = sha256(captchaAppId + captchaAppSecret + ticket + randstr)
```

用服务商分配的 `captchaAppId`、服务商分配的 `captchaAppSecret`，前端验证返回的 `ticket` 和 `randstr`，按上述顺序拼接成字符串，再进行 `sha256` 哈希得到。如下：

```
String captchaAppId = "124324343";
String captchaAppSecret = "xyzxy21121rewrer12yuyu1zxyz";
String ticket = "xyzxy21121rewrer12yuyu1zxyz23123xyzxy21121rewrer12yuyu1zx321321";
String randstr = "das1";
String str = captchaAppId + captchaAppSecret + ticket + randstr;
String sign = sha256(str);
```

验证成功返回样例

```
{
  "code": 200,
  "msg": "成功",
  "taskNo": "987272522132093927999832"
}
```

验证失败返回样例

```
{
  "code": 301,
  "msg": "验证码不正确",
  "taskNo": "987272522132093927999832"
}
```

错误返回

```
{
  "code": 400,
  "msg": "ticket 不能为空"
}
```

返回字段说明

字段名	说明
code	返回码，详见： code 返回码说明
msg	code 对应的描述
taskNo	本次请求号
data	返回具体结果，object 类型，详见 data 返回字段描述

code 返回码说明

code	说明
200	成功

code	说明
301	验证码不正确
302	验证码超时，需要重新获取
303	用户来源 ip 不匹配
400	参数错误
404	接口地址不正确
500	系统维护，请稍后再试
601	接口未开通
602	账号停用
604	接口停用
606	调用超限，请稍后再试
607	ip 不在白名单
609	请求过于频繁，请稍后再试
610	请求超时
999	其他，以实际返回为准