验证码

客户端 SDK 文档

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

客户端 SDK 文档
- Web 前端接入
- Android SDK 接入
- iOS SDK 接入
- 小程序接入

版权所有：腾讯云计算（北京）有限责任公司

客户端 SDK 文档
Web 前端接入

最近更新时间：2019-11-07 14:28:15

前提条件

接入验证码前，需要先在 验证码控制台 中注册 AppID 和 AppSecret。注册完成后，您可以在控制台的 基础配置 中查看 AppID 以及 AppSecret。

接入步骤

快速接入

以下为 Web 端快速接入流程，适用于每次都需要进行人机验证的场景（登录、注册、下发短信、活动等）。

1. 在 Head 标签的最后加入以下代码，引入验证 JS 文件（建议直接在 HTML 中引入）。

```
<script src="https://ssl.captcha.qq.com/TCaptcha.js"></script>
```

2. 在您需要激活验证码的 DOM 元素（button、div、span）内加入以下 ID 及属性。

```
<!--点击此元素会自动激活验证码-->
<!--id：元素的 ID (必须)-->
<!--data-appid：AppID(必须)-->
<!--data-cbfn：回调函数名(必须)-->
<!--data-biz-state：业务自定义透传参数(可选)-->
<button id="TencentCaptcha"
data-appid="appId"
data-cbfn="callback"
type="button"
>验证</button>
```

3. 为验证码创建回调函数。

> 注意：
> 函数名要与 data-cbfn 相同。

```
window.callback = function(res){
console.log(res)
// res（用户主动关闭验证码）= {ret: 2, ticket: null}
// res（验证成功） = {ret: 0, ticket: "String", randstr: "String"}
if(res.ret === 0){
alert(res.ticket) // 票据
}
}
```

完成以上操作后，单击激活验证码的 DOM 元素，即可弹出验证码。至此，验证码客户端接入已完成，您可以进行后台 API 接入 操作。

定制接入

验证码会在全局注册一个 TencentCaptcha 类，业务方可以使用 TencentCaptcha 类自行初始化验证码，并对验证码进行显示或者隐藏操作。
默认的验证码的 JS（TCaptcha.js）在加载完成后，会检测页面中是否存在 `id="TencentCaptcha"` 的元素，如果存在，则会自动将验证码的触发事件绑定在该元素上。如不希望默认绑定，请避免使用 `id="TencentCaptcha"` 的元素。

- **构造函数**
  TencentCaptcha 支持多种参数的重载，以下3种初始化方法，可根据具体情况选择其中一种。

  i. 手动初始化。

  ```
  new TencentCaptcha(appId, callback, options);
  ```

  参数说明：
  - appId：String，申请的场景 ID。
  - callback：Function，回调函数。
  - options：Object，更多配置参数, 请参见 配置参数。

  > 注意：
  > 手动初始化的情况，一般是单击一个元素，执行一段逻辑，才调用验证码，绑定单击的元素不要使用 `id="TencentCaptcha"` 的元素，避免重复绑定单击。

  ii. 绑定到一个元素。

  ```
  new TencentCaptcha(element);
  ```

参数说明：

element： HTMLElement，验证码将绑定 click 事件到该元素上，该方式需要确保元素上有 data-appid 和 data-cbfn 属性。

> 注意：
> 手动绑定不要使用 `id="TencentCaptcha"` 的元素，避免重复绑定单击。

iii. 手动初始化并绑定到一个元素。

```
new TencentCaptcha(element, appId, callback, options);
```

参数说明：

- element: HTMLElement, 需要绑定 `click` 事件的元素

> 注意：
> 手动绑定不要使用 `id="TencentCaptcha"` 的元素，避免重复绑定单击。

- appId: String，申请的场景 ID。
- callback: Function， 回调函数。
- options: Object，更多配置参数, 请参见 配置参数。

- **示例代码**

```
//方法1: 直接生成一个验证码对象。
var captcha1 = new TencentCaptcha('appId', function(res) {/* callback */});
captcha1.show(); // 显示验证码

//方法2:绑定一个元素并自动识别场景id和回调。
// 验证码会读取dom上的`data-appid`和`data-cbfn`以及`data-biz-state`(可选)自动初始化
new TencentCaptcha(document.getElementById('TencentCaptchaBtn'));

//方法3：绑定一个元素并手动传入场景Id和回调。
new TencentCaptcha(
document.getElementById('TencentCaptchaBtn'),
'appId',
function(res) {/* callback */},
{ bizState: '自定义透传参数' }
);
```

回调内容

前端验证成功后，验证码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。结果字段说明如下：

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。
appid	String	场景 ID。
bizState	Any	自定义透传参数。
randstr	String	本次验证的随机串，请求后台接口时需带上。

实例方法

TencentCaptcha 的实例提供一些操作验证码的常用方法：

方法名	说明	传入参数	返回内容
show	显示验证码。	无	无
destroy	隐藏验证码。	无	无
getTicket	获取验证码验证成功后的 ticket。	无	`Object:{"appid":"","ticket":""}`

> 说明：
> show 与 destroy 可以反复调用。

配置参数

options 提供以下配置参数：

配置名	值类型	说明
bizState	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 callback 回调的对象中。

Android SDK 接入

最近更新时间：2019-10-29 11:55:06

前提条件

准备 AppID

验证码接入前，需要先在 验证码控制台 中注册 AppID 和 AppSecret，注册完成后，您可以在控制台的 基础配置 中查看 AppID 以及 AppSecret。

SDK 包下载与运行环境准备

- 单击 下载 Android SDK。
- 本 SDK 运行环境与项目要求：适用于 Android4.0 及以上的系统版本。SDK 工具包目录结构说明如下：

```
.
├── sdk # captchasdk-release-1.0.2.aar
└── CaptchaDemo # 示例工程,演示了如何使用captchasdk-release-1.0.2.aar
```

接入步骤

本 SDK 提供验证码 Dialog 和验证码 Activity 两种使用方式。

方式1：使用验证码 Dialog

1. 创建一个验证码 dialog。

```
/**
@param context，上下文
@param appid，业务申请接入验证码时分配的appid
@param listener，验证码验证结果回调
@param jsonString，业务自定义参数
*/
TCaptchaDialog dialog = new TCaptchaDialog(context, appid, listener, jsonString);
```

2. 显示验证码。

```
dialog.show();
```

其中 jsonString 是 json 字符串，可以为 null。

```
JSONObject jsonObject = new JSONObject();
jsonObject.put("uin", Integer.parseInt(uin));
jsonString = URLEncoder.encode(jsonObject.toString(), "utf-8");
```

其中 listener 根据回调函数返回的 json 对象值判断验证是否成功。

```
/**
json 对象值：jsonObject.getInt("ret")
*/
TCaptchaVerifyListener listener = new TCaptchaVerifyListener() {
@Override
public void onVerifyCallback(JSONObject jsonObject) {
int ret = jsonObject.getInt("ret");
if(ret == 0) {
//验证成功回调
//jsonObject.getInt("ticket")为验证码票据
//jsonObject.getString("appid")为 AppID
//jsonObject.getString("randstr")为随机串
} else if(ret == -1001) {
//验证码首个 TCaptcha.js 加载错误，业务可以根据需要重试
//jsonObject.getString("info")为错误信息
} else {
//验证失败回调，一般为用户关闭验证码弹框
}
}
}
```

方式2：使用验证码 Activity

1. 创建 intent。

```
/**
@param AppID，业务申请接入验证码时分配的 AppID
*/
Intent intent = new Intent(this, TCaptchaPopupActivity.class);
intent.putExtra("appid", ***);
/**
用户自定义参数，可选；
-自定义参数放到 intent 的 map 字段中
JSONObject jsonObject = new JSONObject();
jsonObject.put("uin", Integer.parseInt(uin));
intent.putExtra("map", URLEncoder.encode(jsonObject.toString(), "utf-8"));
*/
startActivityForResult(intent, 1);
```

版权所有：腾讯云计算（北京）有限责任公司

2. 传入 AppID，启动 TCaptchaPopupActivity。

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
if(requestCode == 1 && data != null) {
switch(resultCode) {
case Activity.RESULT_OK: {
JSONObject jsonObject = new JSONObject(data.getStringExtra("retJson"));
int ret = jsonObject.getInt("ret");
if(ret == 0) {
//验证成功回调，此时ret=0，
//jsonObject.getInt("ticket")为验证码票据
//jsonObject.getString("appid")为appid
//jsonObject.getString("randstr")为随机串
} else if(ret == -1001) {
//验证码收个TCaptcha.js加载错误，业务可以根据需要重试
//jsonObject.getString("info")为错误信息
} else {
//验证失败回调，一般为用户关闭验证码弹框
}
break;
}
case Activity.RESULT_CANCELED: {
//用户按了返回键，关闭验证码未验证成功
break;
}
default:
break;
}
}
}
```

至此，验证码客户端接入已完成，您可以进行 后台 API 接入 操作。

iOS SDK 接入

最近更新时间：2019-08-20 10:34:30

前提条件

准备 AppID

验证码接入前，需要先在 验证码控制台 中注册 AppID 和 AppSecret，注册完成后，您可以在控制台的 基础配置 中查看 AppID 以及 AppSecret。

SDK 包下载与运行环境准备

- 单击 下载 iOS SDK。
- 本 SDK 运行环境与项目要求：适用于 iOS8 及以上的系统版本。SDK 工具包目录结构说明如下：

```
.
├── sdk # TCWebCodesSDK.framework
└── demo # 示例工程,演示了如何使用 TCWebCodesSDK.framework
```

接入步骤

1. 在工程中导入 SDK 库：

```
TCWebCodesSDK.framework
```

2. 在需要加载验证码的地方，引入头文件：

```
#import <TCWebCodesSDK/TCWebCodesBridge.h>
```

3. 接入验证码。

```
/**
@param appid，在验证码接入平台注册申请；
@param callback，为回调函数，验证码验证完成后回调该函数将结果带回
*/
[[TCWebCodesBridge sharedBridge]loadTencentCaptcha:self.view appid:@"***" callback:^(NSDictionary *resultJSON) {
if(0==[resultJSON[@"ret"] intValue]) {
/**
验证成功
返回内容：
```

```
resultJSON[@"appid"]为回传的业务appid
resultJSON[@"ticket"]为验证码票据
resultJSON[@"randstr"]为随机串
*/
} else {
/**
验证失败
返回内容：
ret=-1001为验证码js加载错误
ret=-1002一般为网络错误
ret=-1为返回票据数据解析错误，业务可根据需要重试处理
ret的其他返回值，为验证失败，例如用户主动关闭了验证码弹框
*/
}
}];
```

至此，验证码客户端接入已完成，您可以继续完成 后台 API 接入 。

API 解析

1. 返回单例。

```
//返回单例
+ (instancetype)sharedBridge;
```

2. 加载 H5 验证码。

```
/**
@param view，需要加载验证码的视图
@param appid，业务申请接入验证码时分配的appid
@param callback，验证码验证结果回调， 成功/失败可以通过 resultJSON[@"ret"] 判断，0为成功，非0为失败
*/
- (void) loadTencentCaptcha:(UIView*)view appid:(NSString*)appid callback:(void (^)(NSDictionary *resultJSON))callback;
```

3. 设置验证码的显示位置。

```
/**
@note，该接口为可选接口，需在loadTencentCaptcha之前调用。不调该接口时，验证码的中心点将默认为<loadTencentCaptcha>中view的中心位置
@param center，验证码的中心点坐标
```

```
 */
- (void)setCaptchaPosition:(CGPoint)center;
```

示例：

```
[[TCWebCodesBridge sharedBridge] setCaptchaPosition:CGPointMake(self.view.bounds.size.width
*0.5, self.view.bounds.size.height*0.4)];
[[TCWebCodesBridge sharedBridge] loadTencentCaptcha:self.view appid:@"**** callback:^(NSDicti
onary *resultJSON) {
[self showResultJson:resultJSON];
}];
```

4. 设置 NSLog 开关。

```
/**
默认开关为关闭状态，打开后可在控制台输出一些验证码加载过程或出错信息
@note，该接口为可选接口，一般在接入遇到问题时打开用于辅助调试。需在loadTencentCaptcha之前调
用
@param enable，开关状态，YES时打开，NO为关闭
*/
- (void)setLogState:(BOOL)enable;
```

示例：

```
[[TCWebCodesBridge sharedBridge] setLogState:YES;
[[TCWebCodesBridge sharedBridge] loadTencentCaptcha:self.view appid:@"**** callback:^(NSDicti
onary *resultJSON) {
[self showResultJson:resultJSON];
}];
```

小程序接入

最近更新时间：2019-08-20 10:34:37

前提条件

验证码接入前，需要先在 验证码控制台 中注册 AppID 和 AppSecret，注册完成后，您可以在控制台的 基础配置 中查看 AppID 以及 AppSecret。

接入步骤

1. 小程序唤起验证码

通过 wx.navigateToMiniProgram 跳转至验证码小程序，用户完成或关闭验证码后，将返回至调用方的小程序。

在 app.json 配置 navigateToMiniProgramAppIdList，如下：

```
{
"navigateToMiniProgramAppIdList": ["wx5a3a7366fd07e119"]
}
```

wxml 如下：

```
<button bindtap="toTCaptcha">验证</button>
```

js 如下：

```
toTCaptcha: function () {
wx.navigateToMiniProgram({
appId: 'wx5a3a7366fd07e119',
path: '/pages/captcha/index',
extraData: {
appId: 'appId'//您申请的验证码的 appId
}
})
}
```

2. 在 app.js 获取验证结果

验证成功后，验证码小程序页面关闭，验证结果会返回给调用验证码的小程序页面。由于小程序间相互跳转过程中产生的数据仅能在 app.js 中获取到，故需要在 app.js 的 onShow 中添加以下代码，来捕获验证结果 captchaResult。

版权所有：腾讯云计算（北京）有限责任公司

```
App({
// ...
onShow: function(options) {
// 解决各类回调的兼容问题
if (!this.captchaTicketExpire) this.captchaTicketExpire = {};

if (options.scene === 1038 && options.referrerInfo.appId === 'wx5a3a7366fd07e119') {
const result = options.referrerInfo.extraData;
if (result.ret === 0) {
const ticket = result.ticket;
if (!this.captchaTicketExpire[ticket]) {
this.captchaResult = result;
this.captchaTicketExpire[ticket] = true;
}
} else {
// 用户关闭了验证码
}
}
},
// ...
});
```

验证结果（captchaResult）参数说明：

参数名	类型	说明
ret	Number	• 0：验证成功。 • 1：验证失败。 • 2：用户主动关闭验证码。
ticket	String	验证成功的票据，用于后台校验验证合法性，仅验证成功时会生成，
randstr	String	用于后台校验验证票据的随机字符串，仅验证成功时会生成。

3. 将验证结果返回至服务端校验

在小程序页面的 onShow 阶段，将验证结果及待提交的表单数据一起提交到服务器，进行校验。

```
// page.js
const app = getApp()

Page({
data: {
// ...
},
```

```
onShow() {
const captchaResult = app.captchaResult;
app.captchaResult = null; // 验证码的票据为一次性票据，取完需要置空
if (captchaResult && captchaResult.ret === 0) {
// 将验证码的结果返回至服务端校验
// const ticket = captchaResult.ticket;
// const randstr = captchaResult.randstr;
}
},
// ...
});
```

至此，验证码客户端接入已完成，您可以进行 后台 API 接入 操作。