Captcha

Business Disaster Recovery Scheme

Product Documentation

©2013-2024 Tencent Cloud. All rights reserved.

Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice

Tencent Cloud

All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

©2013-2024 Tencent Cloud. All rights reserved.

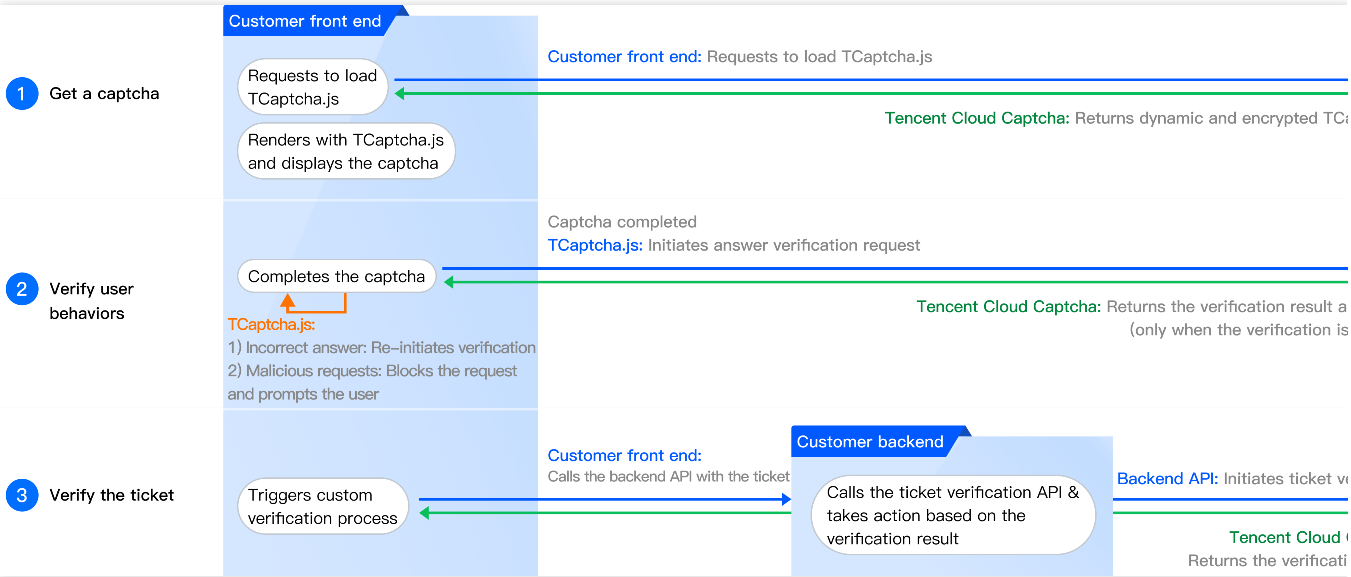
Business Disaster Recovery Scheme

Business Disaster Recovery Scheme (Web and App)

Last updated：2024-11-06 11:38:17

Business request process

The CAPTCHA request process involves the interaction between the business client (frontend), business server (backend), and Captcha server. The sequence diagram of calls is shown as follows:


Customer front end
1 Get a captcha
Requests to load TCaptcha.js
Renders with TCaptcha.js and displays the captcha
Customer front end: Requests to load TCaptcha.js
Tencent Cloud Captcha: Returns dynamic and encrypted TC
2 Verify user behaviors
Completes the captcha
TCaptcha.js:
1) Incorrect answer: Re–initiates verification
2) Malicious requests: Blocks the request and prompts the user
Captcha completed
TCaptcha.js: Initiates answer verification request
Tencent Cloud Captcha: Returns the verification result a
(only when the verification is
3 Verify the ticket
Triggers custom verification process
Customer front end:
Calls the backend API with the ticket
Customer backend
Calls the ticket verification API & takes action based on the verification result
Backend API: Initiates ticket v
Tencent Cloud
Returns the verificati

To ensure that customers' business processes run normally in case of an exception in the Captcha server, we provide the following business disaster recovery schemes.

Note:

No billing is incurred when verifying any disaster recovery tickets generated in the disaster recovery schemes.

Business client (frontend) disaster recovery

1. Define the JS load error handling function.

```
// The error handling function ensures that event processes run normally in case of
// Define the function before the script loads
function loadErrorCallback() {
    var appid = 'CaptchaAppId';
    // Generate a disaster recovery ticket or use another handling technique
    var ticket = 'terror_1001_' + appid + '_' + Math.floor(new Date().getTime() / 1
    callback({
        ret: 0,
        randstr: '@'+ Math.random().toString(36).substr(2),
        ticket,
        errorCode: 1001,
        errorMessage: 'jsload_error',
    });
}
```

2. Call the JS load error handling function if an error is caught during the CAPTCHA instance call.

```
try {
    // Generate a CAPTCHA object
    var captcha = new TencentCaptcha(document.getElementById('cap_iframe'), 'Your C
    // Call the method to show the CAPTCHA
    captcha.show();
} catch (error) {
    // Load error. Call the CAPTCHA js load error handling function.
    loadErrorCallback();
}
```

3. Define the CAPTCHA callback function so the handling is based on ticket and errorCode (instead of ret). For errorCode definitions, please see Web Frontend Integration.

```
function callback(res) {
    // res (CAPTCHA is closed by the user) = {ret: 2, ticket: null}
    // res (Verification is successful) = {ret: 0, ticket: "String", randstr: "Stri
    // res (Request error. A disaster recovery ticket with the prefix terror_ is re
    if (res.ticket){
        // Handle based on errorCode
        if(res.errorCode === xxxxx){
           // Customize the disaster recovery logic (for example, skip this verific
        }
      }
}
```

Business server (backend) disaster recovery

In case of an exception when requesting the **ticket verification API**, the business server needs to handle the exception (for example, skip this verification) to avoid affecting business processes due to abnormal API responses, request timeouts, or the service not responding. **The following are some abnormal responses that require disaster recovery on the business side.**

©2013-2024 Tencent Cloud. All rights reserved.

Request timeout or service not responding.

Exception returned. Code is InternalError. Example:

```
{
  "Response": {
      "Error": {
          "Code": "InternalError",
          "Message": "An internal error has occurred. Retry your request. If the pr
      },
      "RequestId": "xxxxxxxxxxx"
  }
}
```

Service internal error. CaptchaCode is 26. Example:

```
{
  "Response": {
      "CaptchaCode": 26,
      "CaptchaMsg": "System busy. For more information, please see the TenDI Captch
      "EvilLevel": 0,
      "GetCaptchaTime": 0,
      "RequestId": "xxxxxxxxxxx"
  },
  "retcode": 0,
  "retmsg": "ok"
}
```

More information

You can log in to the Captcha console, and click **Quick Consulting** in the upper right corner to learn more.