

title: 验证-ios-sensebot-api type: 验证-ios-sensebot-api order: 0

fold: true

Protocol

GT3CaptchaButtonDelegate

captchaButtonShouldBeginCaptcha:

控制是否执行验证事件

Declaration

```
- (BOOL)captchaButtonShouldBeginCaptcha:(GT3CaptchaButton *)button;
```

Parameters

Param	Description
button	验证按钮

Discussion

默认返回 YES , 只有当返回 NO 时不执行验证事件

captchaButton:didChangeState:

验证状态改变的通知回调

Declaration

```
- (void)captchaButton:(GT3CaptchaButton *)button didChangeState:(GT3CaptchaButtonState)state;
```

Parameters

Param	Description
button	验证按钮
state	当前的按钮状态

GT3CaptchaManagerDelegate

gtCaptcha:errorHandler:

内部错误处理

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager errorHandler:(GT3Error *)error;
```

Parameters

Param	Description
manager	验证管理器
error	错误对象

gtCaptcha:didReceiveSecondaryCaptchaData:response:error:decisionHandler:

通知已经收到二次验证结果, 并请在此处理最终验证结果

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager didReceiveSecondaryCaptchaData:(NSData *)data response:(NSURLResponse *)response error:(GT3Error *)error decisionHandler:(void (^)(GT3SecondaryCaptchaPolicy captchaPolicy))decisionHandler;
```

Parameters

Param	Description
manager	验证管理器
data	二次验证返回的数据
response	二次验证的响应
error	错误对象
decisionHandler	更新验证结果的视图

shouldUseDefaultRegisterAPI:

将要向**API1**发送请求的时候调用此方法

Declaration

```
- (BOOL)shouldUseDefaultRegisterAPI:(GT3CaptchaManager *)manager;
```

Parameters

Param	Description
manager	验证管理器

Return Value

返回是否使用默认的 `API1` 行为

gtCaptcha:willSendRequestAPI1:withReplacedHandler:

将要向**API1**发送请求的时候调用此方法, 通过此方法可以修改将要发送的请求

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager willSendRequestAPI1:(NSURLRequest *)originalRequest withReplacedHandler:(void (^)(NSURLRequest * request))replacedHandler;
```

Parameters

Param	Description
manager	验证管理器
requestHandler	修改请求的执行block

gtCaptcha:didReceiveDataFromAPI1:withError:

将要向API1发送请求的时候调用此方法, 通过此方法可以修改将要发送的请求

Declaration

```
- (NSDictionary *)gtCaptcha:(GT3CaptchaManager *)manager didReceiveDataFromAPI1:(NSDictionary *)dictionary withError:(GT3Error *)error;
```

Parameters

Param	Description
manager	验证管理器
dictionary	包含极验的验证数据
error	返回的错误

- 参数 `dictionary` 内数据样例

```
{
  "challenge" : "12ae1159ffdfcbbc306897e8d9bf6d06",
  "gt" : "ad872a4e1a51888967bdb7cb45589605",
  "success" : 1
}
```

Return Value

返回自定义解析后的 `challenge` , `gt` , `success` 数据, 结构参考上方样例

gtCaptcha:didReceiveCaptchaCode:result:message:

通知接收到返回的验证交互结果

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager didReceiveCaptchaCode:(NSString *)code result:(NSDictionary *)result message:(NSString *)message;
```

Parameters

Param	Description
manager	验证管理器
code	验证交互结果
result	二次验证数据
message	附带消息

Discussion

此方法仅仅是前端返回的初步结果, 并非验证最终结果。获得 `result` 后还需进一步二次验证, 以校验数据是否伪造。

shouldUseDefaultSecondaryValidate:

将要向API2发送请求的时候调用此方法。

Declaration

```
- (BOOL)shouldUseDefaultSecondaryValidate:(GT3CaptchaManager *)manager;
```

Parameters

Param	Description
manager	验证管理器

Return Value

默认返回 YES, 表示 manager 使用二次验证默认逻辑。

gtCaptcha:willSendSecondaryCaptchaRequest:withReplacedRequest:

通知接收到返回的验证交互结果

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager willSendSecondaryCaptchaRequest:(NSURLRequest *)originalRequest withReplacedRequest:(void (^)(NSMutableURLRequest * request))replacedRequest
```

Parameters

Param	Description
manager	验证管理器
requestHandler	修改二次验证请求的block

Discussion

请不要修改**requestHandler**执行所在的线程或队列, 否则可能导致请求修改失败. 二次验证的请求方式应为**POST**, 头部信息应为:

```
{ "Content-Type":@"application/x-www-form-urlencoded;charset=UTF-8" }
```

gtCaptchaUserDidCloseGTVView:

用户主动关闭了验证码界面

Declaration

```
- (void)gtCaptchaUserDidCloseGTVView:(GT3CaptchaManager *)manager;
```

Parameters

Param	Description
manager	验证管理器

GT3CaptchaManagerViewDelegate

gtCaptcha:notifyCaptchaMode:

通知验证模式

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager notifyCaptchaMode:(GT3CaptchaMode)mode;
```

Parameters

Param	Description
manager	验证管理器
mode	验证模式

gtCaptchaWillShowGTView:

通知图形界面将要显示

Declaration

```
- (void)gtCaptchaWillShowGTView:(GT3CaptchaManager *)manager;
```

Parameters

Param	Description
manager	验证管理器

gtCaptcha:updateCaptchaStatus:

更新验证状态

Declaration

```
- (void)gtCaptchaWillShowGTView:(GT3CaptchaManager *)manager;
```

Parameters

Param	Description
manager	验证管理器
state	验证状态

gtCaptcha:updateCaptchaViewWithFactor:to:timeInterval:

更新验证视图

Declaration

```
- (void)gtCaptcha:(GT3CaptchaManager *)manager updateCaptchaViewWithFactor:(CGFloat)fromValue to:(CGFloat)toValue timeInterval:(NSTimeInterval)timeInterval;
```

Parameters

Param	Description
manager	验证管理器
fromValue	起始值
toValue	终止值
timeInterval	时间间隔

GT3CaptchaManagerStatisticDelegate

略, 请阅读头文件注释

GT3AsyncTaskProtocol

executeRegisterTaskWithCompletion:

用于自定义验证注册的任务

Declaration

```
- (void)executeRegisterTaskWithCompletion:(void (^)(GT3RegisterParameter *
_Nullable params, GT3Error * _Nullable error))completion;
```

Parameters

Param	Description
completion	将 API1 处理的结果返回给管理器

Seealso

[GT3RegisterParameter](#)

executeValidationTaskWithValidateParam:completion:

用于自定义验证结果校验的任务

Declaration

```
- (void)executeValidationTaskWithValidateParam:(GT3ValidationParam *)param
completion:(void (^)(BOOL validationResult, GT3Error * _Nullable error))
completion;
```


Parameters

Param	Description
param	图形验证的结果
completion	将 API2 处理的结果返回给管理器

Seealso

[GT3ValidationParam](#)

GT3CaptchaButton

极验提供的验证按钮, 继承于 [UIControl](#)

Property

captchaManager

验证管理器

Declaration

```
@property (nonatomic, readonly, strong) GT3CaptchaManager *captchaManager;
```

Seealso

[GT3CaptchaManager](#)

delegate

验证按钮代理

Declaration

```
@property (nonatomic, weak) id<GT3CaptchaButtonDelegate> delegate;
```

captchaState

验证状态

Declaration

```
@property (nonatomic, readonly, assign) GT3CaptchaState captchaState;
```

captchaEdgeInsets

定义容器视图边距

Declaration

```
@property (nonatomic, assign) UIEdgeInsets captchaEdgeInsets;
```

tipsDict

定义各种验证状态下按钮上的提示文案

Declaration

```
@property (nonatomic, strong) NSDictionary<NSString *, NSAttributedString*> *tipsDict;
```

Discussion

字典请使用以下键名, 与 `GT3CaptchaState` 一一对应

```
'inactive', 'active', 'initial', 'waiting', 'collecting', 'computing', 'success', 'fail', 'error'.
```

Seealso

`GT3CaptchaState`

mainColor

The background color of CAPTCHA widget.

Declaration

```
@property (nonatomic, strong) UIColor *mainColor;
```

borderWidth

Define width for CAPTCHA widget border. Defaults to 1.0. Animatable.

Declaration

```
@property (nonatomic, assign) CGFloat *borderWidth;
```

cornerRadius

Define CAPTCHA widget corner radius. Defaults to 3.0. Animatable.

Declaration

```
@property (nonatomic, assign) CGFloat *cornerRadius;
```

indicatorColor

定义验证状态指示器的颜色

Declaration

```
@property (nonatomic, strong) UIColor *indicatorColor;
```

borderColor

Define color for CAPTCHA widget border. Defaults to 0xcccccc. Animatable.

Declaration

```
@property (nonatomic, strong) UIColor *borderColor;
```

Method

initWithFrame:captchaManager:

初始化并返回一个新的规定了尺寸的 `GT3CaptchaButton` 实例对象

Declaration

```
-(instancetype)initWithFrame:(CGRect)frame captchaManager:(GT3CaptchaManager *)captchaManager;
```

Parameters

Param	Description
frame	按钮的尺寸
captchaManager	验证管理器的实例

Return Value

一个新的规定了尺寸的 `GT3CaptchaButton` 实例对象

startCaptcha

开始验证

Declaration

```
- (void)startCaptcha;
```

Discussion

根据验证状态, 在 `GT3CaptchaManager` 内部调用实例方法 `startGT3CaptchaWithAnimated:`, `requestGT3Captcha`, `showGTViewIfRegistered`。

stopCaptcha

终止验证

Declaration

```
- (void)stopCaptcha;
```

resetCaptcha

重置验证

Declaration

```
- (void)resetCaptcha;
```

updateTitleLabel:

立即更新当前的验证提示标题

Declaration

```
- (void)updateTitleLabel:(NSAttributedString *)title;
```

Parameters

Param	Description
title	提示标题

GT3CaptchaManager

Property

delegate

验证管理的代理方法

Declaration

```
@property (nonatomic, weak) id<GT3CaptchaManagerDelegate> delegate;
```

viewDelegate

验证视图代理

Declaration

```
@property (nonatomic, weak) id<GT3CaptchaManagerViewDelegate> viewDelegate;
```

statisticDelegate

验证统计代理

Declaration

```
@property (nonatomic, weak) id<GT3CaptchaManagerStatisticDelegate> statisticDelegate;
```

captchaState

验证状态

Declaration

```
@property (nonatomic, readonly) GT3CaptchaState captchaState;
```

isShowing

图形验证的展示状态

Declaration

```
@property (nonatomic, readonly) BOOL isShowing;
```

API_1

获取启动验证参数的接口

Declaration

```
@property (nonatomic, readonly) NSURL *API_1;
```

API_2

进行二次验证的接口

Declaration

```
@property (nonatomic, readonly) NSURL *API_2;
```

gt_captcha_id

本次验证会话的验证ID

Declaration

```
@property (nonatomic, readonly, strong) NSString *gt_captcha_id;
```

gt_challenge

本次验证的会话的流水号

Declaration

```
@property (nonatomic, readonly, strong) NSString *gt_challenge;
```

gt_success_code

当前验证的服务状态

Declaration

```
@property (nonatomic, readonly, strong) NSNumber *gt_success_code;
```

Discussion

1正常/0宕机

maskColor

验证背景颜色

Declaration

```
@property (nonatomic, strong) UIColor *maskColor;
```

Method

sdkVersion

SDK版本号

Declaration

```
+ (NSString *)sdkVersion;
```

sharedGTManager

验证单例

Declaration

```
+ (instancetype)sharedGTManager;
```

initWithAPI1:API2:timeout:

验证初始化方法

Declaration

```
- (instancetype)initWithAPI1:(NSString *)api_1  
                        API2:(NSString *)api_2  
                        timeout:(NSTimeInterval)timeout NS_DESIGNATED_INITIALIZER
```

Parameters

Param	Description
api_1	获取验证参数的接口
api_2	进行二次验证的接口
timeout	超时时长

Return Value

一个新的GT3CaptchaManager实例

Discussion

`NS_DESIGNATED_INITIALIZER` , 请不要使用 `init`

cancelRequest

取消异步请求

Declaration

```
- (void)cancelRequest;
```

Discussion

当希望取消正在执行的`NSURLSessionDataTask`时，调用此方法

configureGTest:challenge:success:withAPI2:

自定义配置验证方法

Declaration

```
- (void)configureGTest:(NSString *)gt_public_key
                  challenge:(NSString *)gt_challenge
                  success:(NSNumber *)gt_success_code
                  withAPI2:(NSString *)api_2;
```

Parameters

Param	Description
gt_public_key	自定义时需要实现的动画block,仅在type配置为GTIndicatorCustomType时才执行
gt_challenge	状态指示器的类型
gt_success_code	网站主服务器监测geetest服务的可用状态, 0/1 不可用/可用
api_2	用于二次验证的接口.网站主根据极验服务端sdk来部署

Discussion

该方法已抛弃。请使用 `registerCaptchaWithCustomAsyncTask:completion:` 来自定义 API1 和 API2 请求流程。

同一个challenge只能使用在同一次验证会话中，如果使用该方法配置验证参数，需要开发者自己控制对 `startGTCaptchaWithAnimated:` 的调用，以避免在一次验证会话中多次访问该方法。

registerCaptcha:

注册验证

Declaration

```
- (void)registerCaptcha:(GT3CaptchaDefaultBlock)completionHandler;
```

Parameters

Param	Description
completionHandler	注册完成的回调

Discussion

必须在 `startGTCaptchaWithAnimated:` 前调用

Seealso

[GT3CaptchaDefaultBlock](#)

registerCaptchaWithCustomAsyncTask:completion:

注册验证，并且自定义 API1 及 API2 流程

Declaration

```
- (void)registerCaptchaWithCustomAsyncTask:(id<GT3AsyncTaskProtocol>)customAsyncTask completion:(GT3CaptchaDefaultBlock)completionHandler;
```

Parameters

Param	Description
customAsyncTask	自定义 API1 及 API2 任务对象
completionHandler	注册完成的回调

Discussion

必须在 `startGTCaptchaWithAnimated:` 前调用

Seealso

[GT3CaptchaDefaultBlock](#) 和 [GT3AsyncTaskProtocol](#)

startGTCaptchaWithAnimated:

开始验证

Declaration

```
- (void)startGTCaptchaWithAnimated:(BOOL)animated;
```

Parameters

Param	Description
animated	控制验证视图弹出动画的启动

Discussion

获取姿态, 提交分析后, 如有必要在keyWindow上创建极验验证的GTView验证视图

极验验证GTWebView通过JS与SDK通信

stopGTCaptcha

终止验证

Declaration

```
- (void)stopGTCaptcha;
```

resetGTCaptcha

重置验证

Declaration

```
- (void)resetGTCaptcha;
```

Discussion

内部实现是先调用 `stopGTCaptcha` , 在主线程延迟0.3秒后, 执行 `startCaptcha` 方法, 且只有在 `captchaState` 为 `GT3CaptchaStateFail` , `GT3CaptchaStateError` , `GT3CaptchaStateSuccess` , `GT3CaptchaStateCancel` 才会有效。

执行完后, 状态为 `GT3CaptchaStateInit` 。

closeGTViewIfIsOpen

若验证显示则关闭验证界面

Declaration

```
- (void)closeGTViewIfIsOpen;
```

getCookieValue:

获取cookie value

Declaration

```
- (NSString *)getCookieValue:(NSString *)cookieName;
```

Parameters

Param	Description
cookieName	cookie的键名

Discussion

获取姿态, 提交分析后, 如有必要在keyWindow上创建极验验证的GTView验证视图

Return Value

返回cookie value

useGTViewWithTimeout:

配置 GTView 资源请求超时时长

Declaration

```
- (void)useGTViewWithTimeout:(NSTimeInterval)timeout;
```

Parameters

Param	Description
timeout	超时时长

useGTViewWithCornerRadius:

设置图形验证的圆角大小

Declaration

```
- (void)useGTViewWithCornerRadius:(CGFloat)cornerRadius;
```

Parameters

Param	Description
cornerRadius	圆角大小

Discussion

默认 2px。大小不超过 30 px。

useGTViewWithParams:

验证标题

Declaration

```
- (void)useGTViewWithParams:(NSDictionary *)params;
```

Parameters

Param	Description
params	自定义参数

Discussion

内部将参数转换成表单格式，并拼接到静态资源的请求上。

useGTViewWithTitle:

验证标题

Declaration

```
- (void)useGTViewWithTitle:(NSString *)title;
```

Parameters

Param	Description
title	验证标题字符串

Discussion

默认不开启. 字符长度不能超过28, 一个中文字符为两个2字符长度.

useAnimatedAcitvityIndicator:withIndicatorType:

验证视图高度约束


Declaration

```
- (void)useAnimatedAcitvityIndicator:(GT3IndicatorAnimationViewBlock)animationBlock
                                   withIndicatorType:(GT3ActivityIndicatorType)type
```

Parameters

Param	Description
animationBlock	动画执行的block
type	状态指示器类型

Discussion

为了能方便的调试动画,真机调试模拟低速网络, Settings->Developer->Status->Enable->Edge()

useVisualViewWithEffect:

配置背景模糊

Declaration

```
- (void)useVisualViewWithEffect:(UIBlurEffect *)blurEffect;
```

Parameters

Param	Description
blurEffect	模糊效果

Discussion

iOS8以上生效

useLanguage:

切换验证语言

Declaration

```
- (void)useLanguage:(GT3LanguageType) type;
```

Parameters

Param	Description
type	语言类型

Discussion

默认语言跟随系统。

Seealso

可指定语言。详细见 [GT3LanguageType](#)

useLanguageCode:

切换验证语言

Declaration

```
- (void)useLanguageCode:(NSString *)lang;
```

Parameters

Param	Description
lang	语言简码。请参考下方的语言简码列表。

Discussion

默认跟随系统语言。不支持的语言则使用英文。

语种	Language	Key
阿拉伯语	Arabic	ar
德语	German	de
英语	English	en
欧洲西班牙语	Spanish (Europe)	es
法语	French	fr
印尼语	Indonesian	id
日语	Japanese	ja
韩语	Korean	ko
欧洲葡萄牙语	Portuguese (Europe)	pt-PT
俄语	Russian	ru
简体中文	Chinese (Simplified)	zh-CN
香港繁体	Chinese (Hong Kong)	zh-HK
台湾繁体	Chinese (Taiwan)	zh-TW

useServiceNode:

切换验证服务集群节点

Declaration

```
- (void)useServiceNode:(GT3CaptchaServiceNode)node;
```

Parameters

Param	Description
node	集群节点

Discussion

默认中国节点。使用其他节点需要使用相应的配置，否则无法正确访问验证服务。使用该方法前，请充分了解极验的服务集群节点使用方法。

Seealso

详见 [GT3CaptchaServiceNode](#)

disableSecurityAuthentication:

控制使用HTTPS协议请求验证

Declaration

```
- (void)disableSecurityAuthentication:(BOOL)disable;
```

Parameters

Param	Description
disable	是否禁止https支持

Discussion

默认开启HTTPS

disableBackgroundUserInteraction:

控制验证背景交互事件

Declaration

```
- (void)disableBackgroundUserInteraction:(BOOL)disable;
```

Parameters

Param	Description
disable	控制背景的点击交互事件

Discussion

默认不禁止

enableNetworkReachability:

控制内部的网络可达性检测

Declaration

```
- (void)enableNetworkReachability:(BOOL)enable;
```

Parameters

Param	Description
enable	默认不禁止

Discussion

默认不禁止

enableDebugMode:

Debug Mode

Declaration

```
- (void)enableDebugMode:(BOOL)enable;
```

Parameters

Param	Description
disable	控制debug模式

Discussion

默认不开启

GT3RegisterParameter

Property

gt

验证ID(gt)

Declaration

```
@property (nonatomic, strong) NSString *gt;
```

challenge

验证流水号

Declaration

```
@property (nonatomic, strong) NSString *challenge;
```

success

验证当机状态。@(1) 为正常, @(0) 为宕机。

Declaration

```
@property (nonatomic, strong) NSNumber *success;
```

GT3ValidationParam

Property

code

验证初步判定结果。@"1" 通过, @"0" 未通过。

Declaration

```
@property (nonatomic, strong) NSString *code;
```

result

验证结果校验数据。使用该数据, 通过 validate 接口进行结果校验, 以获得最终验证结果。

Declaration

```
@property (nullable, nonatomic, strong) NSDictionary *result;
```

message

附带的消息。

Declaration

```
@property (nullable, nonatomic, strong) NSString *message;
```

GT3Error

极验封装的错误对象, 用于方便构造和返回特别信息

GT3ErrorType

极验定义的错误类型

Declaration

```
typedef NS_ENUM(NSUInteger, GT3ErrorType) {  
    /** 用户中断验证导致 */  
    GT3ErrorTypeUser,  
    /** 服务端返回错误 */  
    GT3ErrorTypeServer,  
    /** 内部网络抛出错误类型 */  
    GT3ErrorTypeNetWorking,  
    /** 内部浏览器抛出的错误类型 */  
    GT3ErrorTypeWebView,  
    /** 从前端库抛出的错误类型 */  
    GT3ErrorTypeJavaScript,  
    /** 内部解码错误类型 */  
    GT3ErrorTypeDecode,  
    /** 未知错误类型 */  
    GT3ErrorTypeUnknown  
};
```

Property

metaData

发生错误时接收到的元数据, 没有数据则为nil

Declaration

```
@property (nonatomic, readonly, strong) NSData * _Nullable metaData;
```

gtDescription

极验的额外错误信息

Declaration

```
@property (nonatomic, readonly, strong) NSString * gtDescription;
```

originalError

原始的错误

Declaration

```
@property (nonatomic, readonly, strong) NSError * _Nullable originalError;
```

Method

errorWithDomainType:code:userInfo:withGTDescription:

通过提供的详细的参数初始化GT3Error

Declaration

```
+ (instancetype)errorWithDomainType:(GT3ErrorType)type code:(NSInteger)code userInfo:(nullable NSDictionary *)dict withGTDescription:(NSString *)description;
```

Parameters

Param	Description
type	极验定义的错误类型
code	错误码
dict	错误的 <code>userInfo</code>
description	错误的额外描述字段

errorWithDomainType:originalError:withGTDescription:

基于提供的NSError封装成GT3Error

Declaration

```
+ (instancetype)errorWithDomainType:(GT3ErrorType)type originalError:(NSError *)originalError withGTDescription:(NSString *)description;
```

Parameters

Param	Description
type	极验定义的错误类型
error	原始的 <code>NSError</code> 实例对象
description	错误的额外描述字段

GT3Utils

极验验证工具类

GT3CaptchaState

极验验证状态的枚举量

Declaration

```
typedef NS_ENUM(NSInteger, GT3CaptchaState) {  
    /** 验证未激活 */  
    GT3CaptchaStateInactive,  
    /** 验证激活 */  
    GT3CaptchaStateActive,  
    /** 验证初始化中 */  
    GT3CaptchaStateInitial,  
    /** 验证检测数据中 */  
    GT3CaptchaStateCollecting,  
    /** 验证等待提交中 */  
    GT3CaptchaStateWaiting,  
    /** 验证结果判定中*/  
    GT3CaptchaStateComputing,  
    /** 验证通过*/  
    GT3CaptchaStateSuccess,  
    /** 验证失败*/  
    GT3CaptchaStateFail,  
    /** 验证发生错误*/  
    GT3CaptchaStateError  
};
```

GT3CaptchaServiceNode

验证集群节点

Declaration

```
typedef NS_ENUM(NSInteger, GT3CaptchaServiceNode) {  
    /** 中国服务集群*/  
    GT3CaptchaServiceNodeCN = 0,  
    /** 北美服务集群*/  
    GT3CaptchaServiceNodeNA,  
    /** 默认服务集群*/  
    GT3CaptchaServiceNodeDefault = GT3CaptchaServiceNodeCN  
};
```

GT3CaptchaMode

验证模式枚举量

Declaration

```
typedef NS_ENUM(NSInteger, GT3CaptchaMode) {  
    /** 验证默认模式*/  
    GT3CaptchaModeDefault,  
    /** 验证宕机模式*/  
    GT3CaptchaModeFailback  
};
```

GT3SecondaryCaptchaPolicy

视图上结果的更新策略

Declaration

```
typedef NS_ENUM(NSInteger, GT3SecondaryCaptchaPolicy) {  
    /** 二次验证通过 */  
    GT3SecondaryCaptchaPolicyAllow,  
    /** 二次验证拒绝 */  
    GT3SecondaryCaptchaPolicyForbidden  
};
```

GT3LanguageType

语言选项

Declaration

```
typedef NS_ENUM(NSInteger, GT3LanguageType) {
    /** Simplified Chinese 简体中文 */
    GT3LANGTYPE_ZH_CN = 0,
    /** Traditional Chinese 繁体中文 */
    GT3LANGTYPE_ZH_TW,
    /** Traditional Chinese 繁体中文 */
    GT3LANGTYPE_ZH_HK,
    /** Korean 韩语 */
    GT3LANGTYPE_KO_KR,
    /** Japanese 日语 */
    GT3LANGTYPE_JA_JP,
    /** English 英语 */
    GT3LANGTYPE_EN,
    /** Indonesian 印度尼西亚语 */
    GT3LANGTYPE_ID,
    /** Arabic 阿拉伯语 */
    GT3LANGTYPE_AR,
    /** German 德语 */
    GT3LANGTYPE_DE,
    /** Spanish 西班牙语 */
    GT3LANGTYPE_ES,
    /** French 法语 */
    GT3LANGTYPE_FR,
    /** Portuguese 葡萄牙语 */
    GT3LANGTYPE_PT_PT,
    /** Russian 俄语 */
    GT3LANGTYPE_RU,
    /** System language 跟随系统语言*/
    GT3LANGTYPE_AUTO = 999
};
```

GT3ActivityIndicatorType

活动指示器类型

Declaration

```
typedef NS_ENUM(NSInteger, GT3ActivityIndicatorType) {
    /** System Indicator Type */
    GT3IndicatorTypeSystem = 0,
    /** Geetest Default Indicator Type */
    GT3IndicatorTypeDefault,
    /** Custom Indicator Type */
    GT3IndicatorTypeCustom,
};
```

Parameters

Param	Description
code	验证结果
result	二次验证的数据
message	其他消息

GT3CaptchaDefaultBlock

验证默认回调

Declaration

```
typedef void(^GT3CaptchaDefaultBlock)(void);
```

GT3IndicatorAnimationViewBlock

自定义状态指示器的动画实现block

Declaration

```
typedef void(^GT3IndicatorAnimationViewBlock)(CALayer *layer, CGSize size, UIColor *color);
```

Parameters

Param	Description
layer	状态指示器视图的layer
size	layer的大小,默认 {64, 64}
color	layer的颜色,默认 蓝色 [UIColor colorWithRed:0.3 green:0.6 blue:0.9 alpha:1]