

ToTokGameSDK API Document

(iOS)

Name	Date	version	remark
Yuanjun	2020.3.18	1.0.0	created

1. Description and definition

1.1 API call

Every API is called by `[ToTokGameManager defaultManager]` instance

1.2 Callback

`error` is a common field for every callback class. if `error` equals to `nil`, it means the api call is success.

1.3 Callback classes:

1.3.1 Login Callback

```
class: TTGCUserCompletionHandler
fields:
userInfo //the user information
error
```

1.3.2 Receive Notification Callback

```
class: TTGCNotificationCompletionHandler  
fields:  
result      //the notification message  
error
```

1.3.3 Logout Callback

```
class: TTGCLogoutCompleteHandler  
fields:  
success      // BOOL type, indicate operation result  
error
```

1.3.4 Get Friends Callback

```
class: TTGCFriendsCompletionHandler  
fields:  
list          //fields list, contains from 0 -> N friend object  
error
```

1.3.5 Share Callback

```
class: TTGCShareCompleteHandler  
fields:  
success      // BOOL type, indicate operation result  
error
```

1.3.6 Payment Callback

```
class: TTGCOrderProgressHandler  
fields:  
TTGCOderStatus //Payment order status
```

1.3.7 Payment Result Callback

```
class: TTGCOrderCompletionHandler
fields:
orderInfo //the payment order object
error
```

1.3.8 Common Callback

```
class: TTGCCompletionHandler
fields:
result //operation result
error
```

1.4 Errors

TTGCPlatformErrorType_Unknown	unkown error
TTGCPlatformErrorType_AuthorizeFailed	authorized failed
TTGCPlatformErrorType_ForUserInfoFailed	request user data failed
TTGCPlatformErrorType_LoginFailed	login failed
TTGCPlatformErrorType_FriendsListFailed	get firends failed
TTGCPlatformErrorType_MessageSendFailed	send message failed
TTGCPlatformErrorType_PayFailed	payment failed
TTGCPlatformErrorType_ShareFailed	share failed
TTGCPlatformErrorType_NotInstall	app is not installed
TTGCPlatformErrorType_NotNetWork	network exception
TTGCPlatformErrorType_SourceError	third error

2. Setup Initialize

Methods in AppDelegate are implement like below:

Setup Configuration:

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

    // set SDK Api Enviroment

    [[ToTokGameManager defaultManager] setServerDomainTest];

    // open log

    [[ToTokGameManager defaultManager] openLogInfo];

    // launch SDK

    NSString *appId = @"xxxxx";
    NSString *appSecret = @"xxxxxxx";
    NSString *gameid = @"xxxxxxx";

    [[ToTokGameManager defaultManager] setupWithAppId:appId Secret:appSecret
GameId:gameid Application:application Options:launchOptions AnimationHandle:^(BOOL
isFinish) {

        //launch animation finished and load your UI

    }];

    return YES;
}
```

External Callback:

```
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url
options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> *)options {

    return [[ToTokGameManager defaultManager] application:app openURL:url
options:options];
}
```

App Events:

```
- (void)applicationDidEnterBackground:(UIApplication *)application {
```

```

        [[ToTokGameManager defaultManager] applicationDidEnterBackground:application];
    }

- (void)applicationWillEnterForeground:(UIApplication *)application {
    [[ToTokGameManager defaultManager] applicationWillEnterForeground:application];
}

- (void)applicationDidBecomeActive:(UIApplication *)application {
    [[ToTokGameManager defaultManager] applicationDidBecomeActive:application];
}

- (void)applicationWillTerminate:(UIApplication *)application {
    [[ToTokGameManager defaultManager] applicationWillTerminate:application];
}

```

Push Register:

```

- (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {
    [[ToTokGameManager defaultManager] registerDeviceToken:deviceToken];
}

- (void)application:(UIApplication *)application
didReceiveRemoteNotification:(NSDictionary *)userInfo fetchCompletionHandler:(void
(^)(UIBackgroundFetchResult))completionHandler {
    [[ToTokGameManager defaultManager] application:application
didReceiveRemoteNotification:userInfo];
}

```

2. API: Configuration

2.1 Enbale Log Output

```

[[ToTokGameManager defaultManager] openLogInfo];

```

2.2 Disable Log Output(default)

```

[[ToTokGameManager defaultManager] closeLogInfo];

```

2.3 Set Test Environment

```
[[ToTokGameManager defaultManager] setServerDomainTest];
```

2.4 Set Production Environment(default)

```
[[ToTokGameManager defaultManager] setServerDomainProduction];
```

2.5 Set Sandbox Payment

```
[[ToTokGameManager defaultManager] setPayEnvironmentTest];
```

2.6 Set Production Payment(default)

```
[[ToTokGameManager defaultManager] setPayEnvironmentTest];
```

3. API: Login&User Info

User must agree the User Agreement & Privacy before do Login action

3.1 User Agreement Url

```
NSString *urlString = [[ToTokGameManager defaultManager] getAgreementUrlString];
```

3.2 Privacy Url

```
NSString *urlString = [[ToTokGameManager defaultManager] getPrivacyUrlString];
```

3.3 Login with GameCenter

Method	loginWithGameCenterCompletion
Callback	TTGCUserCompletionHandler

example:

```
- (void)gamecenterLogin {
    TTGCHUD_NO_Stop(@"login...")

    __weak __typeof(self)weakSelf = self;

    [[ToTokGameManager defaultManager] loginWithGameCenterCompletion:^(id _Nonnull
userInfo, NSError * _Nonnull error) {
        if (!error) {
            //login success

            TTGCHUD_HINT(@"success")

            [weakSelf closeLoginView];
        } else {
            //error info

            TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);
        }
    }];
}
```

3.4 Login with Facebook

Method	loginWithFacebookCompletion
Callback	TTGCUserCompletionHandler

example:

```
- (void)facebookLogin {
    TTGCHUD_NO_Stop(@"login...")

    __weak __typeof(self)weakSelf = self;

    [[ToTokGameManager defaultManager] loginWithFacebookCompletion:^(id _Nonnull
userInfo, NSError * _Nonnull error) {
```

```

        if (!error) {

            //login success

            TTGCHUD_HINT(@"success")

            [weakSelf closeLoginView];

        } else {

            //error info

            TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);

        }

    }

}];
}

```

3.5 Login with Guest

Method	guestLoginCompletion
Callback	TTGCUserCompletionHandler

example

```

- (void)guestLogin {

    TTGCHUD_NO_Stop(@"login...")

    __weak __typeof(self)weakSelf = self;

    [[ToTokGameManager defaultManager] guestLoginCompletion:^(id _Nonnull userInfo,
NSError * _Nonnull error) {

        if (!error) {

            //login success

            TTGCHUD_HINT(@"success")

            [weakSelf closeLoginView];

        } else {

            //error info

            TTGCHUD_Stop

        }

    }

}];
}

```


3.6 Get User Info

Method	userInfoCompletion
Callback	TTGCUUserCompletionHandler
Params	<div>TTGCUUserModel</div> <div><div>userType</div><div>user type (TTGCLoginType by 3.7)</div></div> <div><div>userId</div><div>user's indentifier</div></div> <div><div>nickname</div><div>nickname</div></div> <div><div>photoUrl</div><div>head image url</div></div> <div><div>pushOff</div><div>push switch 0:push switch on 1:push switch off</div></div>

example

```
- (void)getUserInfo {
    TTGCHUD_NO_Stop(@"")
    [[ToTokGameManager defaultManager] userInfoCompletion:^(id _Nullable userInfo,
    NSError * _Nullable error) {
        if (!error) {
            TTGCHUD_HINT(@"success")
            TTGCUUserModel *model = userInfo;
            [self freshUserInfo:model];
        } else {
            //error info
            TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);
        }
    }];
}
```

3.7 Get Login Type

Method	loginType
Return	TTGCLoginType

Definition:

Enum	TTGCLoginType
Description	Enum type for Login
Enum Values	
TTGCLoginType_unloggedIn	not login
TTGCLoginType_ToTok	ToTok user
TTGCLoginType_GameCenter	GameCenter player
TTGCLoginType_Facebook	Facebook user
TTGCLoginType_Guest	ToTokGame guest

example:

```
- (void)showLoginView {

    if ([[ToTokGameManager defaultManager] loginType] == TTGCLoginType_unloggedIn) {

        //unlogged in and show login UI

        [self backgroundUI];

        [self loginView];

    } else {

        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(2.5 * NSEC_PER_SEC)),
dispatch_get_main_queue(), ^{

            [self getUserInfo];

        }));

    }

}
```

4. API: Logout

Method	logout
Callback	TTGCLogoutCompleteHandler

example:

```
- (IBAction)logout:(id) sender {

    TTGCHUD_NO_Stop(@"logout...")

}
```

```

[[ToTokGameManager defaultManager] logout:^(BOOL success, NSError * _Nullable error)
{
    if (success) {
        TTGCHUD_HINT(@"success")

        [self showLoginView];
    } else {
        //error info
        NSLog(@"%@",error.userInfo);

        TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);
    }
}];
}

```

5. API: Game Payment

5.1 Product Purchase

Order Status Definition:

TTGCOderStatus_ProductQuerying	query product
TTGCOderStatus_Generating	order init
TTGCOderStatus_ProductPurchasing	order in trading progress
TTGCOderStatus_ProductPurchased	order purchased
TTGCOderStatus_ReceiptChecking	order verifying

Method	buyProductWithSKU: Progress: Completion:
Params	
sku	String,product id
Progress	TTGCOderProgressHandler

Completion	<pre> TTGOrderCompletionHandler class TTGOrderModel orderId orderRef //transaction_id channel //1 env //0=sandbox, 1=production sku //product id amount //price createTime payTime cancelTime status //0=order created, 1=order payed, 2=order reset </pre>
------------	--

remark:

the Game Server should send the product to user's account after the order be payed, then check the order status by server api, Order status check is necessary for anti-spam

Server API: <ToTokGame Server API Document (to Game Developer)>

example:

```

- (IBAction)pay:(id)sender {

    TTGCHUD_NO_Stop(@"pay...")

    [[ToTokGameManager defaultManager] buyProductWithSKU:@"LangChaoJD001"]

    Progress:^(TTGCOderStatus orderStatus) {

        NSLog(@"order status: %ld", (long)orderStatus);

    } Completion:^(id _Nullable orderInfo, NSError * _Nullable error) {

        if (orderInfo) {

            TTGCHUD_HINT(@"success")

        }

        if (error) {

            NSLog(@"%@", error.userInfo);

            TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);

        }

    }

};

```

```
}
```

5.2 Query Order

Method	queryOrderWithOrderId: Completion:
Params	
orderId	the order's id
Completion: n	TTGCOderCompletionHandler TTGCOderModel by 5.1

example:

```
- (IBAction)query:(id)sender {
    [[ToTokGameManager defaultManager] queryOrderWithOrderId:@"xxx"
    Completion:^(id _Nullable orderInfo, NSError * _Nullable error) {
        if (orderInfo) {
            //get order info
        } else {
            //error message
        }
    }];
}
```

6. API: SNS & Share

6.1 ToTok Friends

Not yet open

6.2 Share

Share to Facebook, WhatsApp.

ToTok Sharing is not yet open.

6.2.1 Share To Facebook

Method	facebookShareMessage: completion:
Params	
Message	1.TTGCSocialFBInvite quote //message text
	2.TTGCSocialFBImages photos //NSArray image urls
completion	TTGCShareCompleteHandler

example:

```
// share invite link to facebook
- (IBAction)fbShare:(id)sender {
    TTGCSocialFBInvite *obj = [[TTGCSocialFBInvite alloc] init];
    obj.quote = @"Join us";
    [[ToTokGameManager defaultManager] facebookShareMessage:obj completion:^(BOOL success,
NSError * _Nullable error) {
        if (success) {
        } else {
            if (error) {
                TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);
            }
        }
    }];
}
```

6.2.2 Share To WhatsApp

Method	whatsAppShareMessage: completion:
Params	

Message	1.TTGCSocialWAInvite contentString // whatsapp message text
	2.TTGCSocialWAImages photos //NSArray image urls
completion	TTGCShareCompleteHandler

example:

```
// share invite link to whatsapp

- (IBAction)whatsAppShare:(id)sender {

    TTGCSocialWAInvite *obj = [[TTGCSocialWAInvite alloc] init];

    obj.contentString = @"play games";

    [[ToTokGameManager defaultManager] whatsAppShareMessage:obj completion:^(BOOL success,
NSError * _Nullable error) {

        if (success) {

        } else {

            if (error) {

                TTGCHUD_HINT([error.userInfo objectForKey:@"errorMsg"]);

            }

        }

    }];

}
```

7. API: Push

7.1 Register

in AppDelegate:

```
- (void)application:(UIApplication *)application
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {

    [[ToTokGameManager defaultManager] registerDeviceToken:deviceToken];

}
```

7.2 Receive Push Message

Method	getRemoteNotification
Callback	TTGCNotificationCompletionHandler

7.3 Clean Push Message

```
[[ToTokGameManager defaultManager] cleanNotification];
```

7.4 Push Setting

Method1	checkSystemNotificationEnable
Method2	systemNotificationSetting

example:

```
- (IBAction)check:(id)sender {  
    [[ToTokGameManager defaultManager] checkSystemNotificationEnable:^(BOOL isOn) {  
        //isOn YES:opened notification    NO:closed notification  
        if (!isOn) {  
            //Jump to page of system notification settings  
            [[ToTokGameManager defaultManager] systemNotificationSetting];  
        } else {  
            TTGCHUD_HINT(@"System notification already opened~")  
        }  
    }  
};  
}
```

8. Universal Link

Development in progress

9. analytic

Reference resources: <ToTokGame Server API Document (to Game Developer)>