

腾讯云维纳斯

SDK手册和发布版本

产品文档



腾讯云

【版权声明】

©2015-2016 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】



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

文档声明.....	2
iOS SDK手册	4
Android SDK手册	16
iOS SDK版本发布	24
Android SDK版本发布.....	27

iOS SDK手册

initWithAppID

```

- (instancetype)initWithAppID:(int)appID
    andAppVersion:(NSString *)appVersion
    andAppChannel:(NSString *)appChannel;
    
```

接口说明：初始化Wns系统

参数说明：

参数	类型	必选	说明
appID	int	是	WNS控制台分配给app的id
appVersion	NSString	否	app的版本号
appChannel	NSString	否	app的发布渠道信息:该值填"DEBUG"和"RDM"时,后台会分别使用DEBUG和RDM证书进行苹果推送,填其它值时,默认使用正式的证书来进行苹果推送

reset

```

- (void)reset:(BOOL)reconnect;
    
```

接口说明：重置wns连接

参数说明：

参数	类型	必选	说明
reconnect	BOOL	是	重置连接方式 YES：先关闭

参数	类型	必选	说明
			现有WNS通道再重新建立 NO：只关闭现有WNS通道

setStatusCallback

```
- (void) setStatusCallback:(WnsStatusCallback) callback;
```

```
typedef void (^WnsStatusCallback) (WnsSDKStatus status);
```

```
typedef NS_ENUM(NSUInteger, WnsSDKStatus) {
    WnsSDKStatusDisconnected    = 0,    //Wns□□□□□□
    WnsSDKStatusConnecting      = 1,    //Wns□□□□□□
    WnsSDKStatusConnected       = 2,    //Wns□□□□□□
};
```

接口说明：获取wns网络状态

参数说明：

参数	类型	必选	说明
callback	WnsStatusCallback	是	在WnsStatusCallback回调 block中同步当前wns通道 状态

setLogCallback

```
- (void) setLogCallback:(WnsLogCallback) callback;
```

```
typedef void (^WnsLogCallback) (NSString *log, WnsSDKLogLevel level);
```

```
typedef NS_ENUM(NSUInteger, WnsSDKLogLevel) {
    WnsSDKLogLevelDisabled      = -1,
```

```

WnsSDKLogLevelError      = 0,
WnsSDKLogLevelWarn      = 1,
WnsSDKLogLevelInfo      = 2,
WnsSDKLogLevelDebug     = 3,
WnsSDKLogLevelMax       = 4,
};

```

接口说明：获取wns sdk记录的本地log信息

参数说明：

参数	类型	必选	说明
callback	WnsLogCallback	是	在WnsLogCallback回调block中获取wns sdk记录在本地的log信息

bind

```
- (void)bind:(NSString *)uid completion:(void (^)(NSError *error))completion;
```

接口说明：绑定开发商用户id到wns系统中

参数说明：

参数	类型	必选	说明
uid	NSString	是	开发商自己的用户id
completion	block	是	调用结果的回调block

unbind

```
- (void)unbind:(NSString *)uid completion:(void (^)(NSError *error))completion;
```

接口说明：从wns系统中解绑开发商用户id

参数说明：

参数	类型	必选	说明
uid	NSString	是	开发商自己的用户id
completion	block	是	调用结果的回调block

sendRequest

```

- (long)sendRequest:(NSData *)data
                 cmd:(NSString *)cmd
             timeout:(unsigned int)timeout
        completion:(void (^)(NSString *cmd, NSData *data, NSError
        *bizError, NSError *wnsError, NSError *wnsSubError))completion;
    
```

接口说明：调用后端接口服务(二进制协议)

返回值：

```

-1 : □□
long : □□□□□□
    
```

参数说明：

参数	类型	必选	说明
cmd	NSString	是	接口命令字
timeout	unsigned int	是	调用超时时间
completion	block	是	调用结果的回调block
completion (cmd)	NSString	是	回调中的命令字
completion (data)	NSData	是	回调中后端返回的数据

参数	类型	必选	说明
completion (bizError)	NSError	是	回调中开发者的返回码
completion (wnsError)	NSError	是	回调中wns的主返回码
completion (wnsSubError)	NSError	是	回调中wns的子返回码

sendHTTPRequest

```

- (long) sendHTTPRequest: (NSURLRequest *) request
                    cmd: (NSString *) cmd
                    timeout: (unsigned int) timeout
                    completion: (void (^) (NSString *cmd, NSURLResponse
* response, NSData* data, NSError* wnsError, NSError *wnsSubError)) completion;
    
```

接口说明：调用后端接口服务(HTTP协议)

返回值：

```

-1 : 失败
long : 成功
    
```

参数说明：

参数	类型	必选	说明
request	NSURLRequest	是	系统NSURLRequest对象
cmd	NSString	是	接口命令字
timeout	unsigned int	是	调用超时时间
completion	block	是	调用结果的回调block
completion (cmd)	NSString	是	回调中的命令字
completion (response)	NSURLResponse	是	回调中后端返回的数据转化为NSURLResponse

参数	类型	必选	说明
completion (data)	NSData	是	回调中后端返回的原始数据
completion (wnsError)	NSError	是	回调中wns的主返回码
completion (wnsSubError)	NSError	是	回调中wns的子返回码

cancelRequest

```
- (void)cancelRequest:(long) seqno;
```

接口说明：取消请求

参数说明：

参数	类型	必选	说明
seqno	long	是	请求序列号，从sendRequest或sendHTTPRequest调用返回的序列号

cancelAllRequest

```
- (void)cancelAllRequest;
```

接口说明：取消所有请求

setPushDataReceiveBlock

```
- (void)setPushDataReceiveBlock:(void (^)(NSString *cmd, NSData *data, NSError *error)) completion;
```

接口说明：设置Wns Push数据的数据接收回调block

参数说明：

参数	类型	必选	说明
completion (cmd)	NSString	是	push数据的的命令字
completion (data)	NSData	是	push的具体数据
completion (error)	NSError	是	异常对象

registerRemoteNotification

```
- (void)registerRemoteNotification:(NSString
*)deviceToken completion:(void(^)(NSError *error))completion;
```

接口说明：在wns系统注册苹果的推送服务所使用的devicetoken

参数说明：

参数	类型	必选	说明
deviceToken	NSString	是	push数据的的命令字
completion (data)	block	是	注册结果回调

setLogLevel

```
- (void)setLogLevel:(WnsSDKLogLevel) level;
```

```
typedef NS_ENUM(NSInteger, WnsSDKLogLevel) {
    WnsSDKLogLevelDisabled    = -1,
    WnsSDKLogLevelError      = 0,
    WnsSDKLogLevelWarn       = 1,
    WnsSDKLogLevelInfo       = 2,
    WnsSDKLogLevelDebug      = 3,
    WnsSDKLogLevelMax        = 4,
};
```

```
};
```

接口说明：设置log级别,数值越大越详细

参数说明：

参数	类型	必选	说明
level	WnsSDKLogLevel	是	具体的log级别

log

```
- (void)log:(WnsSDKLogLevel)level file:(const char *)file func:(const char
*)func line:(int)line module:(NSString*)module EFDict:(NSDictionary
*)extDict msg:(NSString *)fmt, ...;
```

```
typedef NS_ENUM(NSInteger, WnsSDKLogLevel) {
    WnsSDKLogLevelDisabled      = -1,
    WnsSDKLogLevelError         = 0,
    WnsSDKLogLevelWarn          = 1,
    WnsSDKLogLevelInfo          = 2,
    WnsSDKLogLevelDebug         = 3,
    WnsSDKLogLevelMax           = 4,
};
```

接口说明：记录日志信息

参数说明：

参数	类型	必选	说明
level	WnsSDKLogLevel	是	具体的log级别
file	const char *	是	记录log处位置对应的文件名

参数	类型	必选	说明
func	const char *	是	记录log处位置对应的函数名
line	int	是	记录log处位置对应的文件行号
module	NSString	是	记录log处位置对应的模块名
extDict	NSDictionary	是	按k/v格式存储的log信息
fmt	NSString	是	动态参数的log信息

setAutoTestMode

```
- (void)setAutoTestMode:(BOOL)isEnabled;
```

接口说明：设置自动测速模式，测速模式下，开发商自己的接口不用接入wns系统，wns sdk会定时发模拟包到wns后台，这样开发商可以评估正式接入后的效果。

参数说明：

参数	类型	必选	说明
isEnabled	BOOL	是	是否启用

setDebugIP

```
- (void)setDebugIP:(NSString *)ip port:(unsigned short)port;
```

接口说明：使用测试环境，测试环境下终端连接到wns测试环境，wns测试环境也会使用开发商的测试服务器(控制台上配置)

参数说明：

参数	类型	必选	说明

参数	类型	必选	说明
ip	NSString	是	使用固定IP : 183.60.76.15
port	unsigned short	是	可选的端口 : 80或0

reportDebugLog

```
- (void)reportDebugLog:(NSString *)title content:(NSString
*)content beginTime:(NSTimeInterval)beginTime endTime:(NSTimeInterval)endTime;
```

接口说明 : 将wns sdk记录在终端的log和用户指定的内容上报到wns后台(可以在控制台查看上报的log信息)

参数说明 :

参数	类型	必选	说明
title	NSString	是	log上报的标题
content	NSString	是	指定上报的内容, 不需要可为空(sdk会自动再加上本地sdk记录的log一起上报)
beginTime	NSTimeInterval	是	上报从这个时间开始的本地log
endTime	NSTimeInterval	是	这个时间后的log不上报

getWnsCarrierType

```
- (WnsSDKCarrierType)getWnsCarrierType;
```

```
typedef NS_ENUM(NSUInteger, WnsSDKCarrierType) {
    WnsSDKCarrierUnknown      = 0, //□□
    WnsSDKCarrierMobile       = 1, //□□
    WnsSDKCarrierUnicom       = 2, //□□
    WnsSDKCarrierTelecom      = 3, //□□
    WnsSDKCarrierTietong      = 4  //□□
};
```

接口说明：取得运营商信息

返回值：

```
WnsSDKCarrierType : □□□□□
```

wid

```
- (int64_t)wid;
```

接口说明：获取当前用户的wid

返回值：

```
int64_t : □□□wid
```

keyLog

```
- (NSString *)keyLog;
```

接口说明：获取sdk记录的关键步骤信息，和debug log的区别是只记录关键信息，便于理解和快速查看

返回值：

```
NSString : □□□□□
```


Android SDK手册

接口WnsService

public interface WnsService

类说明：

WNS客户端功能接口

所有接口只能在使用WNS服务的client进程调用

WNS不支持多进程同时调用

initAndStartWns

```
void initAndStartWns(Application app,
    com.tencent.wns.client.inte.WnsAppInfo info)
```

接口说明：

初始化APP信息并启动wns

参数:

参数名	类型	必选	说明
app	Application	是	应用的Application类
info	WnsAppInfo	是	Wns应用的基本信息

WnsAppInfo类说明：

成员	类型	说明
appId	int	应用的id
appVersion	String	应用版本
channelId	String	应用渠道号
isQuickVerification	boolean	设置快速验证模式，测速模式下，开发商自己的接口不用接入wns系统，wns sdk会定时发模拟包到wns后台，这样开发商可以评估正式接入后的

成员	类型	说明
		效果。
debugIp	String	debug svr设置

sendRequest

```
int sendRequest(java.lang.String cmd,
    int timeout_ms,
    byte[] buff,
    com.tencent.wns.client.inte.IWnsCallback.WnsTransferCallback callback)
```

接口说明：

透传接口

参数:

参数名	类型	必选	说明
cmd	String	是	命令字
timeout_ms	int	是	超时时间，单位ms
buff	byte[]	是	业务数据
callback	WnsTransferCallback	是	回调透传结果

```
public static interface WnsTransferCallback
{
    public abstract void onTransferFinished(IWnsTransferResult re);
}
```

TransferResult

方法	返回类型	说明
getBizBuffer()	byte[]	获取业务数据

方法	返回类型	说明
getWnsCode()	int	获取wns返回码

cancelRequest

```
void cancelRequest(int seqNo);
```

接口说明：

取消请求

参数名	类型	必选	说明
app	int	是	returned from #sendRequest()

getWnsHttpClient

```
HttpClient getWnsHttpClient()
```

接口说明：

获取WNS的HttpClient对象

i)支持Http协议透明接入

i)对于wns sdk内部失败，错误提示语放在reason phrase字段

i)自定义header请不要使用前缀“wns”，该前缀被预留给wns sdk使用

getWnsHttpUrl

```
java.net.URL getWnsHttpUrl(java.lang.String url) throws java.net.MalformedURLException
```

接口说明：

支持Http协议透明接入

返回:

可以使用HttpURLConnection库提供的功能

setDebugIp

```
void setDebugIp(java.lang.String host, int port)
```

接口说明:

设置测试环境

参数:

参数名	类型	必选	说明
host	String	是	测试环境ip
port	int	是	端口号

bind

```
void bind(java.lang.String uid,  
com.tencent.wns.client.inte.IWnsCallback.WnsBindCallback callback)
```

接口说明:

第三方app登录绑定uid

参数:

参数名	类型	必选	说明
uid	String	是	业务用户的唯一标识
callback	WnsBindCallback	是	bind回调

```
public static interface WnsBindCallback  
{  
    public abstract void onBindFinished(BindResult re);  
}
```

BindResult

方法	返回类型	说明
getBizBuffer()	byte[]	获取业务数据
getWnsCode()	int	获取wns返回码

unbind

```
void unbind(java.lang.String uid,
            com.tencent.wns.client.inte.IWnsCallback.WnsUnbindCallback callback)
```

接口说明:

与 #bind(String, String, String, int, WnsUnbindCallback)对应的登出方法

BindResult

方法	返回类型	说明
getBizBuffer()	byte[]	获取业务数据
getWnsCode()	int	获取wns返回码

```
public static interface WnsBindCallback
{
    public abstract void onBindFinished(BindResult re);
}
```

BindResult

方法	返回类型	说明
getBizBuffer()	byte[]	获取业务数据
getWnsCode()	int	获取wns返回码

getStatus

```
java.lang.String getStatus()
```

接口说明:

获取sdk记录的关键步骤信息，便于理解和快速查看

reset

```
void reset()
```

接口说明:

取消当前未完成的所有请求，比如切换帐号时可以调用

getWid

```
long getWid()
```

接口说明:

获取wid

setBackgroundMode

```
void setBackgroundMode(boolean isBackground)
```

接口说明:

应用在切换前后台时通知WNS设置是否后台模式

WNS启动默认是后台模式，所以该方法一定要使用

后台模式下15分钟后进入省电模式，省电模式下WNS会有一些措施：停止测速、减小心跳频率.....

参数:

参数名	类型	必选	说明
isBackground	boolean	是	true 应用进入后台，false 应用进入前台

reportDebugLog

```
void reportDebugLog(java.lang.String detail,
    long time,
    java.util.HashMap externMap)
```

接口说明:

把wns的log上报到的服务器

参数:

参数名	类型	必选	说明
detail	String	是	简短的描述
time	long	是	日志开始时间,上报time向前24h的日志
externMap	HashMap	是	上报WNS debug日志 日志上传的扩展map

setStatusCallback

setStatusCallback

```
void setStatusCallback(WnsService.WnsSDKStatusListener listener)
```

接口说明 :

设置WNS服务状态通知监听器

参数名	类型	必选	说明
listener	WnsSDKStatusListener	是	监听回调

WNS服务状态更新

```
static interface WnsSDKStatusListener
{
```

```
void onWnsStateUpdate(WnsSDKStatus oldState, WnsSDKStatus newState);  
}
```

枚举WnsSDKStatus：WNS服务状态

状态名	说明
Disconnected	网络不可用
Connecting	网络连接建立中
Connected	网络连接建立成功

iOS SDK版本发布

SDK历史版本及更新日志

版本号：WNS SDK V 3.0.9

[下载](#)

发布时间：2016.12.7

版本说明：

- 1.兼容iOS10下ATS特性
- 2.支持发送https请求

升级指南：1.WnsSDK.framework变更为WnsSDK4Cloud.framework

2.app链接时需要添加系统库：libresolv.tbd、libc++.dylib

版本号：WNS SDK V 2.5.21

[下载](#)

发布时间：2016.9.14

版本说明：

- 1.暂时移除iOS10下ATS特性
 - 2.移除不必要的测速逻辑
-

版本号：WNS SDK V 2.5.20

[下载](#)

发布时间：2016.9.6

版本说明：

- 1.兼容iOS10下ATS特性
 - 2.增加获取log文件路径接口
 - 3.替换localdns接口
 - 4.修复WnsUrlProtocol多线程冲突导致crash
 - 5.替换demo内测试http请求地址
-

版本号：WNS SDK V 2.5.18

[下载](#)

发布时间：2016.7.4

版本说明：

- 1.修正HTTP接口的NSURLResponse的URL为nil的问题

版本号：WNS SDK V 2.5.17

[下载](#)

发布时间：2016.6.30

版本说明：

- 1.支持应用在多个环境下配置不同push证书
- 2.简易接入http协议的应用
- 3.增加PUSH接收稳定性
- 4.增加SDK稳定性
- 5.更换腾讯内部应用接入时用到的备份IP

版本号：WNS SDK V 2.5.16

[下载](#)

发布时间：2016.4.5

版本说明：

- 1.修正应用在后台切换到前台时小几率收不到Push的问题

版本号：WNS SDK V 2.5.15

[下载](#)

发布时间：2016.3.24

版本说明：

- 1.修正内部APP连到错误的后台环境的问题

版本号：WNS SDK V 2.5.14

[下载](#)

发布时间：2016.3.22

版本说明：

- 1.兼容IPV6 NAT64环境
 - 2.支持后台新增的Push flag特性
-

版本号：WNS SDK V 2.5.13

[下载](#)

发布时间：2016.1.15

版本说明：

- 1.修正Bug, 增强SDK稳定性
-

版本号：WNS SDK V 2.5.12

[下载](#)

发布时间：2015.12.17

版本说明：

- 1.支持Xcode7的Bitcode特性
 - 2.增加了一些逻辑的容错处理
-

版本号：WNS SDK V 2.5.11

[下载](#)

发布时间：2015.8.19

版本说明：

- 1.提供关键调试日志拉取
 - 2.提供新的方式发送HTTP请求
 - 3.增加了后端调度逻辑的处理
-

Android SDK版本发布

Android SDK历史版本及更新日志

版本号：WNS SDK V 2.6.0

发布时间：2016.12.14

版本说明：

- 1、支持发送https请求

[下载](#)

版本号：WNS SDK V 2.5.14

发布时间：2016.3.21

版本说明：

- 1、适配Android6.0版本
- 2、解决了在64位机器上的兼容问题
- 3、优化了日志的性能
- 4、优化了网络切换中发送请求的体验
- 5、修复了部分bug

[下载](#)

版本号：WNS SDK V 2.5.12

发布时间：2015.9.30

版本说明：

- 1、提供关键调试日志拉取
- 2、增加了后端调度逻辑的处理
- 3、修复了一些bug

[下载](#)