

腾讯云维纳斯

SDK 文档

产品文档



腾讯云

## 【版权声明】

©2013-2017 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
SDK 文档.....	4
iOS SDK.....	4
Android SDK.....	15
iOS SDK版本发布 .....	23
Android SDK版本发布.....	27

SDK 文档

## iOS SDK

### 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 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通道不可用
    WnsSDKStatusConnecting = 1, //Wns通道建立中
    WnsSDKStatusConnected = 2, //Wns通道建立成功
};
```

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

参数	类型	必选	说明
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：失败

long：请求的序列号

参数说明：

参数	类型	必选	说明
cmd	NSString	是	接口命令字

参数	类型	必选	说明
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
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：请求的序列号

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



参数	类型	必选	说明
completion (response)	NSURLResponse	是	回调中后端返回的数据转化为 NSURLResponse
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(NSUInteger, 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
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

参数	类型	必选	说明
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
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(NSInteger, WnsSDKCarrierType) {
    WnsSDKCarrierUnknown = 0, //未知
    WnsSDKCarrierMobile = 1, //移动
    WnsSDKCarrierUnicom = 2, //联通
    WnsSDKCarrierTelecom = 3, //电信
    WnsSDKCarrierTietong = 4 //铁通
};
```

接口说明：取得运营商信息

返回值：

WnsSDKCarrierType : 运营商信息

wid

- (int64\_t)wid;

接口说明：获取当前用户的 wid

返回值：

int64\_t：用户的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：具体的信息

## Android SDK

### 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

成员	类型	说明
		接入后的效果。
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
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

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 3.1.0

### [下载](#)

发布时间：2016.12.19

版本说明：

- 1.提升 sdk 稳定性
- 

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 NSURLConnection 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.支持应用在不同环境下配置不同 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](#)