



Push SDK PHP 版

服务端用户手册

V2.0

发布日期： 2013年4月2日

百度开发者中心

（版权所有，翻版必究）

目录

第 1 章 PHP SDK 总体介绍3

 1.1 使用前提.....3

 1.2 工具组成.....3

 1.3 一般规则.....3

 1.4 错误码集合.....3

第 2 章 API 介绍4

 2.1 __CONSTRUCT.....4

 2.2 SETAPIKEY.....4

 2.3 SETSECRETKEY.....5

 2.4 SETCURLOPTS5

 2.5 GETREQUESTID.....5

 2.6 QUERYBINDLIST.....6

 2.7 VERIFYBIND.....7

 2.8 FETCHMESSAGE.....8

 2.9 FETCHMESSAGECOUNT10

 2.10 DELETMESSAGE.....11

 2.11 PUSHMESSAGE12

 2.12 SETTAG14

 2.13 FETCHTAG15

 2.14 DELETETAG16

 2.15 QUERYUSERTAGS.....17

 2.16 INITAPPIOSCERT.....18

 2.17 UPDATEAPPIOSCERT19

 2.18 QUERYAPPIOSCERT.....20

 2.19 DELETEAPPIOSCERT21

 2.20 QUERYDEVICETYPE.....22

第1章 PHP SDK 总体介绍

将 Push 服务端的所有操作封装成一个类 Channel，通过对该类的简单初始化，即可调用其内部的各种方法，使用百度 Push 服务。

Channel 提供的方法和服务端 API 对应，是对服务端 REST API 的封装。

1.1 使用前提

支持 curl 的 PHP(5.2 及其以上版本)，编译 PHP 时，加上--with-curl={curl_dir}选项。

1.2 工具组成

PHP SDK 工具包主要由以下部分组成：

- Channel.class.php -- PHP_SDK 脚本，包含对外提供的所有接口
- lib -- 使用到的一些基础公用文件
- sample/sample.php -- 展示如何使用 PHP_SDK 的 demo 文件
- phpsdk 下载

1.3 一般规则

- 所有函数的参数和返回值中如果有中文，必须是 UTF-8 编码；
- 不需要对函数参数进行 urlencode。

1.4 错误码集合

如果用户在调用 SDK 时发生错误，那么错误分成两大类：

- 与服务交互失败产生的错误，比如：SDK 参数不完整、网络错误、服务器的返回不是正确的 json 包导致无法解析等，这类错误的错误码位于 1-100 区间，具体如下：

服务器交互失败错误

错误码	错误信息
1	php sdk error
2	php sdk init error
3	lack param
4	http status is error, and the body returned is not a json string
5	http status is ok, but the body returned is not a json string

- 与服务器交互成功，但服务器返回了非 200 的 HTTP 状态，比如用户权限错误、重复绑定等，具体如下：

HTTP 状态错误

错误码	错误信息
30600	Internal Server Error
30601	Method Not Allowed
30602	Request Params Not Valid
30603	Authentication Failed
30604	Quota Use Up Payment Required
30605	Data Required Not Found
30606	Request Time Expires Timeout

30607	Channel Token Timeout
30608	Bind Relation Not Found
30609	Bind Number Too Many
30610	Duplicate Operation

第2章 API 介绍

2.1 __construct

原型

public function __construct (\$apiKey, \$secretKey, \$arr_curl_opts = array())
--

功能

Channel 类构造函数，用户在使用 SDK 之前，必须 new 一个 Channel 对象。

请求参数

参数名称	类型	是否必需	描述
\$apiKey	string	是	从百度开发者中心获得的 Api Key。
\$secretKey	string	是	从百度开发者中心获得的 Secret Key。
\$arr_curl_opts	array	否	设置 curl 的选项，如果不设置，默认 curl_opts = array (CURLOPT_TIMEOUT => 30, CURLOPT_CONNECTTIMEOUT => 5) 设置的参数内容和限制请参考 php 的 curl 库的 curl_setopt 函数说明。

返回值

无。

示例

```
$channel = new Channel (
    $apiKey, $secretKey,
    array ( CURLOPT_TIMEOUT => 30, CURLOPT_CONNECTTIMEOUT => 5 ) ) ;
```

2.2 setApiKey

原型

public function setApiKey(\$apiKey)

功能

Channel 对象初始化后，可以调用此方法重新设置 apiKey。

请求参数

参数名称	类型	是否必需	描述
\$apiKey	string	是	从百度开发者中心获得的 Api Key。

返回值

无。

示例

```
$channel->setApiKey($apiKey);
```

2.3 setSecretKey

原型

```
public function setSecretKey($secretKey)
```

功能

Channel 对象初始化后，可以调用此方法重新设置 secretKey。

请求参数

参数名称	类型	是否必需	描述
\$ secretKey	string	是	从百度开发者中心获得的 Secret Key。

返回值

无。

示例

```
$channel->setSecretKey ($secretKey);
```

2.4 setCurlOpts

原型

```
public function setCurlOpts($arr_curlOpts)
```

功能

Channel 对象初始化后，可以调用此方法重新设置 curl 的选项。

请求参数

参数名称	类型	是否必需	描述
\$arr_curlopts	array	是	要设置的 curlOpt 数组，参数请参考 PHP 的 curl 库。

返回值

无。

示例

```
$channel->setCurlOpts (array ( CURLOPT_TIMEOUT => 30, CURLOPT_CONNECTTIMEOUT => 5 ) );
```

2.5 getRequestId

原型

```
public function getRequestId()
```

功能

获取上次与服务器交互的 request_id，方便出错时的问题追查；如果上次调用并未与服务器交互就失败返回了，那么该函数返回 0。

请求参数

无。

返回值

类型	描述
int	成功，返回上次与服务器交互返回的 request_id; 失败，0，并未与服务器交互就失败返回了。

示例

```
$channel->getRequestId();
```

2.6 queryBindList

原型

```
public function queryBindList ( $user_id, $optional = NULL)
```

功能

查询用户的绑定列表。

请求参数

参数名称	类型	是否必需	描述
\$user_id	string	是	查询的用户编号。
\$optional	array	否	\$optional 可选参数数组，可以设置的非必须参数的 key 包括： Channel::CHANNEL_ID(指定系统分配的 channel_id，必须为字符串)； Channel::DEVICE_TYPE(指定查询设备类型，必须为整数，可选值为： 1： 浏览器设备； 2： pc 设备； 3： Android 设备； 4： ios 设备； 5： windows phone 设备)； Channel::START(起始页码，必须为整数，默认为 0)； Channel::LIMIT(每页查询数量，必须为整数，默认为 100，可选范围为 1-100)。

示例

```
function test_bindList ( $user_id, $device_type )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $optional [ Channel::DEVICE_TYPE ] = $device_type;
    $ret = $channel->queryBindList ( $user_id, $optional ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' ) ;
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' ) ;
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' ) ;
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!!!'. '\n' ) ;
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_bindList OK!!!!
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [total_num] => 10,
            [amount] => 2,
            [binds] => Array
                (
                    [0] => Array
                        (
                            [channel_id] => 12343321543,
                            [user_id] => 324353215431,
                            [device_id] => 222134231,
                            [device_type] => 2,
                            [device_name] => "移动手机 milestone",
                            [bind_name] => "test",
                            [info] => "info test",
                            [bind_status] => 0,
                            [online_status] => "on",
                            [online_timestamp] => 123473434,
                            [online_expires] => 124583423,
                        )
                    [1] => Array
                        (
                            [channel_id] => 238742398472,
                            [user_id] => 324353215431,
                            [device_id] => 4323212331,
                            [device_type] => 2,
                            [device_name] => "移动手机 defy",
                            [bind_name] => "test",
                            [info] => "info test",
                            [bind_status] => 0,
                            [online_status] => "on",
                            [online_timestamp] => 123473434,
                            [online_expires] => 124583423,
                        )
                )
        )
)
```

2.7 verifyBind

原型

```
function test_bindVerify ( $user_id, $device_type )
```

功能

校验 user_id、channel_id 是否已经绑定。

请求参数

参数名称	类型	是否必需	描述
\$user_id	string	是	查询的用户编号
\$optional	array	否	\$optional 可选参数数组，可以设置的非必须参数的 key 包括： Channel::CHANNEL_ID(指定系统分配的 channel_id，必须为字符串)

			Channel::DEVICE_TYPE(指定查询设备类型, 必须为整数, 可选值为: 1: 浏览器设备; 2: pc 设备; 3: Android 设备; 4: ios 设备; 5: windows phone 设备);
--	--	--	---

示例

```
function test_bindVerify ( $user_id, $device_type )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $optional [ Channel::DEVICE_TYPE ] = $device_type;    //Optional
    $ret = $channel->verifyBind ( $user_id, $optional ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' ) ;
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' ) ;
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' ) ;
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' ) ;
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_bindList OK!!!!
result: Array
(
    [request_id] => 4153581200
)
```

2.8 fetchMessage

原型

public function fetchMessage (\$user_id, \$optional = NULL)

功能

根据 user_id, channel_id 查询离线消息。

请求参数

参数名称	类型	是否必需	描述
\$user_id	string	是	查询的用户编号
\$optional	array	否	\$optional 可选参数数组, 可以设置的非必须参数的 key 包括: Channel::CHANNEL_ID(指定系统分配的 channel_id, 必须为字符串) Channel::START (起始页码, 必须为整数, 默认为 0) Channel::LIMIT(每页查询数量, 必须为整数, 默认为 10, 可选范围为 1-10)

示例


```
function test_fetchMessage ( $user_id, $start, $limit )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $optional[Channel::START] = $start;
    $optional[Channel::LIMIT] = $limit;
    $ret = $channel->fetchMessage ( $user_id, $optional );
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_fetchMessage OK!!!!
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [total_num] => 2,
            [messages] => Array
                (
                    [0] => Array
                        (
                            [channel_id] => 12343321543,
                            [msg_id] => 324353215431,
                            [msg_key] => "testkey1",
                            [data] => "hello world",
                            [length] => 11
                            [type] => 0,
                            [expired] => 123948382,
                        )
                    [1] => Array
                        (
                            [channel_id] => 12343321543,
                            [msg_id] => 324353215431,
                            [msg_key] => "testkey2",
                            [data] => "hello world",
                            [length] => 11
                            [type] => 0,
                            [expired] => 123948382,
                        )
                )
        )
)
```

)

2.9 fetchMessageCount

原型

public function fetchMessageCount (\$user_id, \$optional = NULL)

功能

根据 user_id、channel_id 查询离线消息的个数。

请求参数

参数名称	类型	是否必需	描述
\$user_id	string	是	查询的用户编号
\$optional	array	否	\$optional 可选参数数组，可以设置的非必须参数的 key 包括：Channel::CHANNEL_ID(指定系统分配的 channel_id，必须为字符串)

示例

```
function test_fetchMessageCount ( $user_id )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->fetchMessageCount ( $user_id );
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_fetchMessageCount OK!!!!
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [total_num] => 10,
        )
)
```

2.10 deleteMessage

原型

public function deleteMessage (\$user_id, \$msg_ids, \$optional = NULL)

功能

根据 user_id、msg_ids、channel_id 删除离线消息。

请求参数

参数名称	类型	是否必需	描述
\$user_id	string	是	查询的用户编号。
\$msg_ids	array 或 json string	是	要删除消息 id 列表，多个参数则用数组格式，或是由数组转换成的 json 字符串。注意：消息 id 由系统自动生成，与 msg_key 无关。如：123 或[123,124]
\$optional	array	否	\$optional 可选参数数组，可以设置的非必须参数的 key 包括：Channel::CHANNEL_ID(指定系统分配的 channel_id，必须为字符串)

示例

```
function test_deleteMessage ( $user_id, $msg_ids )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->deleteMessage ( $user_id, $msg_ids ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_deleteMessage OK!!!!
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [success_amount] => 2,
            [details] => Array
                (
                    [0] => Array
                        (
```

```
[msg_id] => 12343321543,
[result] => 0
)
[1] => Array
(
    [msg_id] => 123432321543,
    [result] => 0
)
)
)
```

2.11 pushMessage

原型

```
public function pushMessage (
    $push_type,
    $messages,
    $message_keys,
    $optional = NULL )
```

功能

根据 push_type, messages,message_keys 推送消息。

请求参数

参数名称	类型	是否必需	描述
\$ push_type	uint	是	推送类型，取值范围 1~3 1: 单个人，必须指定 user_id 和 channel_id（指定设备上的指定用户）或者 user_id（指定用户的所有设备） 2: 一群人，必须指定 tag_name 3: 所有人，可不指定 tag_name、user_id、channel_id
\$messages	string, array 或 json string	是	要发送的消息列表， 单条消息 messages 可以是字符串,多条消息应为数组格式，或是由数组转换成的 json 字符串。 如果有二进制的消息内容，请先做 BASE64 的编码。一次推送最多 10 个消息。 特别提醒：如果是推送至 ios 或 windows phone 设备的消息，则必须由开发者自行组织消息格式。
\$message_keys	string, array 或 json string	是	Channel::MSG_KEYS 要发送的消息 keys。单个参数可以是字符串， 多个参数应为数组格式，或是由数组转换成的 json 字符串，而且 message_keys 里的内容要和 messages 的内容一一对应。注意：具有相同 message_keys 的消息会自动覆盖。
user_id	string	否	Channel::USER_ID 用户 id，在 Android 里， channel_id + userid 指定某一个特定 client 不超过 256 字节 如 果存在此字段，则只推送给此用户
tag_name	string	否	Channel::TAG_NAME 消息标签名，不超过 128 字

			节
\$channel_id	uint	否	Channel::CHANNEL_ID(指定系统分配的 channel_id, 必须为字符串);
device_type	uint	否	Channel::DEVICE_TYPE(指定查询设备类型, 必须为整数, 可选值为: 1: 浏览器设备; 2: pc 设备; 3: Android 设备; 4: ios 设备; 5: windows phone 设备);
message_type	uint	否	Channel::MESSAGE_TYPE 消息类型 0: 消息 (透传) 1: 通知 默认为 0
message_expires	uint	否	Channel::MESSAGE_EXPIRES 指定消息的过期时间, 默认 86400 秒. 必须和 messages 一一对应。

示例

```
function test_pushMessage ( $messages, $message_keys)
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key) ;
    //推送一条单播消息, 必须指定 user_id 或者 user_id+channel_id
    $push_type = 1;
    $user_id = 'xxx';
    $channel_id = 'xxx';
    $optional[Channel::USER_ID] = $user_id;
    $optional[Channel::CHANNEL_ID] = $channel_id;
    $message = 'Hello World';
    $message_key = 'msg_key';
    $ret = $channel->pushMessage ( $push_type, $message, $message_key,
    $optional );

    //推送多条单播消息, 多条消息的 $messages 与 $message_keys 用数组, 或是由数组转换
    成的 json 字符串表示,
    $messages = array("Hello World" , "Hello Push");
    $message_keys = array("key1" , "key2");
    $ret = $channel->pushMessage ( $push_type, $messages, $message_keys,
    $optional);

    //推送到一群人, 按 tag 推送, 必须指定 tag_name
    $push_type = 2;
    $tag_name = 'xxx';
    $optional[Channel::TAG_NAME] = $tag_name;
    $ret = $channel->pushMessage($push_type, $messages, $message_keys,
    $optional);

    //推送到某个应用下的所有人, 不用指定 user_id, channel_id, tag_name
    $push_type = 3;
    $ret = $channel->pushMessage($push_type, $messages, $message_keys);
    //检查返回值
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
}
```

```
else
{
    echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
    echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
}
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_pushMessage OK!!!!
result: Array
(
    [request_id] => 4153581200
)
```

2.12 setTag

原型

public function setTag (\$tag_name, \$optional = NULL)

功能

创建消息广播组。

必选参数

参数名称	类型	是否必需	描述
\$tag_name	string	是	消息标签名称，不超过 128 字节。
\$optional	array	否	\$optional 可选参数数组，可以设置的非必须参数的 key 包括： Channel:: USER_ID 用户 Id，当 user_id 被提交时，服务端将会完成用户和 tag 的绑定操作

示例

```
function test_setTag( $tag_name, $user_id )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $optional[Channel:: USER_ID] = $user_id;
    $ret = $channel->setTag( $tag_name, $optional );
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息，包括标签 id
bool	失败返回 false

返回值示例

```
SUCC, test_setTag OK!!!!
result: Array
(
    [request_id] => 4153581200,
    [response_params] => Array
        (
            [tid] => 2
        )
)
```

2.13 fetchTag

原型

public function fetchTag (\$optional = NULL)

功能

查询消息标签信息。

请求参数

参数名称	类型	是否必需	描述
\$optional	array	否	\$optional 可选参数数组，可以设置的非必须参数的 key Channel::TAG_NAME 查询的标签名，若被指定，则获取该标签的详细信息。否则，获取按页获取应用的所有标签。 Channel::START (起始页码，必须为整数，默认为 0); Channel::LIMIT(每页查询数量，必须为整数，默认为 10，可选范围为 1-10).

示例

```
function test_fetchTag( )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->fetchTag( );
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_fetchTag OK!!!!
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [tag_num] => 1,
            [tags] => Array
                (
                    [0] => Array
                        (
                            [tid] => 123,
                            [name] => "test",
                            [info] => "test info",
                            [create_time] => 12345678,
                        )
                )
        )
)
```

2.14 deleteTag

原型

public function deleteTag (\$tag_name, \$optional = NULL)

功能

删除消息组。

请求参数

参数名称	类型	是否必需	描述
\$tag_name	String	是	查询的标签编号，必须为整数
\$optional	Array	否	\$optional 可选参数数组，可以设置的非必须参数的 key Channel::USER_ID, 用户 Id, 当 USER_ID 提交时，服务器会解除该用户与 tag 的绑定关系

请求示例

```
function test_deleteTag( $tag_name )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->deleteTag( $tag_name ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {

```



```
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
Array	服务器返回的结果信息
Bool	失败返回 false

返回值示例

```
SUCC, test_deleteTag OK!!!!
result: Array
(
    [request_id] => 4153581200
)
```

2.15 queryUserTags

原型

public function queryUserTags (\$user_id, \$optional = NULL)

功能

查询属于用户的组列表。

请求参数

参数名称	类型	是否必需	描述
\$user_id	string	是	查询的用户编号,不超过 256 字节。

示例

```
function test_queryUserTags( $user_id )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->queryUserTags( $user_id );
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' );
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' );
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_queryUserTag OK!!!!
```

```
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [tag_num] => 1,
            [tags] => Array
                (
                    [0] => Array
                        (
                            [tid] => 123,
                            [name] => "test",
                            [info] => "test info",
                            [create_time] => 12345678,
                        )
                )
        )
)
```

2.16 initApploscert

原型

```
public function initApploscert (
    $name,
    $description,
    $release_cert,
    $dev_cert,
    $optional = NULL )
```

功能

初始化推送到 iOS 设备的应用证书。

请求参数

参数名称	类型	是否必需	描述
\$name	string	是	证书名称，最长 128 字节。
\$description	string	是	证书描述，最长 256 字节。
\$release_cert	string	是	正式版证书内容。
\$dev_cert	string	是	开发版证书内容。

请求示例

```
function test_initApploscert ( $name, $description, $release_cert,
$dev_cert )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->initApploscert ( $name, $description, $release_cert,
$dev_cert ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' ) ;
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' ) ;
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' ) ;
    }
}
```

```
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n');
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' ) ;
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_initAppIoscert OK!!!!
result: Array
(
    [request_id] => 4153581200
)
```

2.17 updateAppIoscert

原型

```
public function updateAppIoscert ( $optional = NULL )
```

功能

修改推送到 ios 设备的应用证书。

请求参数

参数名称	类型	是否必需	描述
\$optional	array	否	optional 可选参数数组，可以设置的非必须参数的 key Channel::NAME 证书名称，最长 128 字节； Channel::DESCRIPTION，证书的描述； Channel::RELEASE_CERT，正式版证书内容 Channel::DEV_CERT，开发版证书内容

请求示例

```
function test_updateAppIoscert ( $name, $description, $release_cert,
$dev_cert )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $optional[ Channel::NAME ] = $name;
    $optional[ Channel::DESCRIPTION ] = $description;
    $optional[ Channel::RELEASE_CERT ] = $release_cert;
    $optional[ Channel::DEV_CERT ] = $dev_cert;
    $ret = $channel->updateAppIoscert ( $optional ) ;
    if ( false == $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' ) ;
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' ) ;
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' ) ;
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n');
    }
}
```

```
else
{
    echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' ) ;
    echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
}
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_updateAppIoscert OK!!!!
result: Array
(
    [request_id] => 4153581200
)
```

2.18 queryAppIoscert

原型

public function queryAppIoscert (\$optional = NULL)
--

功能

查询推送到 ios 设备的应用证书。

请求参数

无。

请求示例

```
function test_queryAppIoscert ( )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->queryAppIoscert ( ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' ) ;
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' ) ;
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' ) ;
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' ) ;
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_queryAppIoscert OK!!!!
result: Array
(
    [request_id] => 4153581200,
    [response_params] => Array
        (
            [name] => "test",
            [description] => "test desc",
            [release_cert] => "XXXXXXXXXXXXXXXXXX",
            [dev_cert] => "XXXXXXXXXXXXXXXXXXXXXXXXXX",
        )
)
```

2.19 deleteAppIoscert

原型

public function deleteAppIoscert (\$optional = NULL)

功能

删除推送到 ios 设备的应用证书。

请求参数

无。

请求示例

```
function test_deleteAppIoscert ( )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->deleteAppIoscert ( ) ;
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' ) ;
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' ) ;
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' ) ;
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' ) ;
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_deleteAppIoscert OK!!!!
result: Array
(
    [request_id] => 4153581200
)
```

2.20 queryDeviceType

原型

```
public function queryDeviceType ( $channel_id, $optional = NULL)
```

功能

根据 Channel ID 查询设备类型。

请求参数

参数名称	类型	是否必需	描述
\$channel_id	uint	是	Channel 的编号

请求示例

```
function test_queryDeviceType ( $channel_id )
{
    global $api_key;
    global $secret_key;
    $channel = new Channel ( $api_key, $secret_key ) ;
    $ret = $channel->queryDeviceType($channel_id);
    if ( false === $ret )
    {
        echo ( 'WRONG, ' . __FUNCTION__ . ' ERROR!!!!\n' );
        echo ( 'ERROR NUMBER: ' . $channel->errno ( ) . '\n' );
        echo ( 'ERROR MESSAGE: ' . $channel->errmsg ( ) . '\n' );
        echo ( 'REQUEST ID: ' . $channel->getRequestId ( ) . '\n' );
    }
    else
    {
        echo ( 'SUCC, ' . __FUNCTION__ . ' OK!!!!'. '\n' ) ;
        echo ( 'result: ' . print_r ( $ret, true ) . '\n' ) ;
    }
}
```

返回值

类型	描述
array	服务器返回的结果信息
bool	失败返回 false

返回值示例

```
SUCC, test_queryDeviceType OK!!!!
result: Array
(
    [request_id] => 4153581200
    [response_params] => Array
        (
            [device_type] => 1, //1:浏览器设备;2:PC 设备;3:Android 设备;4:iOS 设备;5:Windows Phone 设备
        )
)
```