**SUBMAIL C# SDK**

[**概览**](javascript:void(0))

支持NET版本：4.5 以上  
代码编码格式：utf-8  
引入方式：直接将SDK中的Submaill.dll 和Newtonsoft.Json.dll导入

**依赖**

依赖第三方库Newtonsoft.Json.dll

[**文件目录索引**](javascript:void(0))

**Submail**

**AppConfig**

IAppConfig.cs 配置接口文件

MailAppConfig.cs 邮件配置文件

MessageConfig.cs 短信配置文件

VoiceAppConfig.cs 语音验证配置文件

**Lib**

AddressBookMail.cs 邮件地址薄，订阅和添加联系人到目标地薄

AddressBookMessage.cs 短信地址薄，订阅和添加联系人到目标地薄

ISender.cs 发送接口

SendBase.cs Message Send 和Mail Send的基类

Mail.cs 邮件API

Message.cs 短信API

MailSend.cs mail/send

MailXSend.cs mail/xsend

MessageXSend.cs message/xsend

MessageMultiSend.cs message/multi-xsend

VoiceSend.cs 语音验证API

**Utility**

HttpWebHelper.cs 处理HTTP请求

RequestHelper.cs 处理请求数据

SignatureHelper.cs 处理请求数据

**SubMailTest**

AddressBookMailDemo.cs 邮件地址薄测试程序

AddressBookMessageDemo.cs 短信地址薄测试程序

MailSendDemo.cs 邮件send测试程序

MailSendXDemo.cs 短信XSend测试程序

MessageMultiXSendDemo.cs 短信多方发送测试程序

MessageSendXDemo.cs 短信发送测试程序

VoiceSendDemo.cs 语音验证测试程序

|  |
| --- |
|  |
|  |

[**开始使用**](javascript:void(0))

**SDK CLASS索引**

**MailSend mail/send API**

**MailXSend mail/xsend API**

**MessageXSend message/xsend API**

**MessageMultiXSend message/multixsend API**

**AddressbookMail 邮件地址簿 API**

**AddressbookMessage 短信地址簿 API**

**VoiceVerify 语音验证API**

[**MailSend 类**](javascript:void(0))

**初始化**

**IAppConfig mailConfig = new MailAppConfig("AppID","AppKey");**

**MailSend submail = new MailSend(mailConfig);**

**使用指引**

|  |  |
| --- | --- |
| **方法** | **描述** |
| **AddTo:name:** | **添加邮件地址到 To 数组，第一个必选参数：邮件地址。第二个可选参数：收件人姓名** |
| **AddAddressBook:** | **添加地址薄标识到 addressbook 数组** |
| **SetSender:name:** | **设置发件人，第一个必选参数：邮件地址。第二个可选参数：显示名称** |
| **SetReply:** | **设置回复地址** |
| **AddCc:name:** | **添加抄送地址** |
| **AddBcc:name:** | **添加密送地址** |
| **SetSubject:** | **设置邮件标题** |
| **SetText:** | **设置文本邮件内容** |
| **SetHtml:** | **设置 HTML 邮件内容** |
| **AddVar** | **添加文本变量到 vars 数组** |
| **AddLink** | **添加超链接变量到 links 数组** |
| **AddHeaders** | **添加自定义邮件头指令到 headers 数组** |
| **AddAttachment:** | **添加附件到 attachments 数组** |
| **Send:** | **发送邮件** |

**代码示例**

使用 MailSend 类提交 mail/send 发送一封简单的邮件

MailAppConfig("AppID", "AppKey");

MailSend submail = new MailSend(mailConfig);

submail.AddTo("youremail@XXX.com", "yourname");

submail.AddCc("leo@submail.cn", "leo");

submail.AddBcc("leo@submail.cn", "leo");

submail.SetSender("leo@inside.submail.me", "leo");

submail.SetReply("service@submail.cn");

submail.SetSubject("发送历史与明细");

submail.SetText("发送历史与明细");

submail.AddAttachment(@"C:\attachment.txt");

string resultMessage = string.Empty

if (submail.Send(out resultMessage)== false)

{

Console.WriteLine(resultMessage);

}

**MailXSend 类**

**初始化类**

**IAppConfig mailConfig = new MailAppConfig("AppID","AppKey");**

**MailXSend submail = new MailXSend(mailConfig);**

**使用指引**

**方法 描述**

**AddTo 添加邮件地址到 To 数组，第一个必选参数：邮件地址。第二个可选参数：收件人姓名**

**AddAddressbook 添加地址薄标识到 addressbook 数组**

**SetSender 设置发件人，第一个必选参数：邮件地址。第二个可选参数：显示名称**

**SetReply 设置回复地址**

**AddCc 添加抄送地址**

**AddBcc 添加密送地址**

**SetSubject 设置邮件标题**

**SetProject 设置邮件项目标识**

**AddVar 添加文本变量到 vars 数组**

**AddLink 添加超链接变量到 links 数组**

**AddHeaders 添加自定义邮件头指令到 headers 数组**

**GetSender() 获取发送方式**

**XSend 发送邮件**

**代码示例**

使用 MAILXsend 类提交 mail/xsend 发送一封邮件。

IAppConfig mailConfig = new MailAppConfig("AppId", "AppKey");

MailXSend submail = new MailXSend(mailConfig);

submail.AddTo("youremail@mail.com", "yourname");

submail.SetSender("leo@inside.submail.me", "leo");

submail.SetProject("7pfhZ3");

submail.AddHeaders("X-Accept", "zh-cn");

submail.AddHeaders("X-Mailer", "leo App");

string resultMessge = string.Empty;

if(submail.XSend(out resultMessge) == false)

{

Console.WriteLine(resultMessge);

}

**MessageXSend 类**

**初始化类**

**IAppConfig messageConfig= new MessageConfig("AppID","AppKey");**

**MessageXSend submail = new MessageXSend(messageConfig);**

**使用指引**

**方法 描述**

**AddTo 添加11位手机号码**

**AddAddressBook 添加地址薄标识到 addressbook**

**SetProject 设置短信项目标记**

**AddVar 添加文本变量到 vars 数组**

**GetSender 获取发送方式**

**XSend 发送短信**

**代码示例**

使用 MESSAGEXsend 类提交 message/xsend 发送一条触发短信

IAppConfig appConfig = new MessageConfig("AppId", "AppKey");

MessageXSend messageXSend = new MessageXSend(appConfig);

messageXSend.AddTo("your phone number");

messageXSend.SetProject("MApf82");

messageXSend.AddVar("code", "你好");

string returnMessage = string.Empty;

if (messageXSend.XSend(out returnMessage) == false)

{

Console.WriteLine(returnMessage);

}

**AddressBookMail 类**

**初始化类**

**IAppConfig mailConfig = new MailAppConfig("AppID","AppKey");**

**AddressBookMail addressBookMail = new AddressBookMail(mailConfig);**

**使用指引**

**方法 描述**

**SetAddress 设置邮件地址，第一个必选参数：邮件地址。第二个可选参数：收件人姓名**

**SetAddressbook 设置目标地址薄标识**

**Subscribe 订阅或添加联系人**

**UnSubscribe 退订或移除联系人**

**GetSender() 获取发送方式**

**代码示例**

使用 ADDRESSBOOKMail 类提交 addressbook/mail/subscribe 请求，添加订阅联系人或向目标地址薄中添加联系人

IAppConfig mailConfig = new MailAppConfig("AppID","AppKey");

AddressBookMail addressBookMail = new AddressBookMail(mailConfig);

addressBookMail.SetAddress("leo12@submail.cn", "leo");

addressBookMail.SetAddressBook("61RgC3");

string outMessage = string.Empty;

bool issuccess = addressBookMail.Subscribe(out outMessage);

if (!issuccess)

{

Console.WriteLine(outMessage);

}

使用 ADDRESSBOOKMail 类提交 addressbook/mail/unsubscribe 请求

IAppConfig mailConfig = new MailAppConfig("AppID","AppKey");

AddressBookMail addressBookMail = new AddressBookMail(mailConfig);

addressBookMail.SetAddress("leo12@submail.cn", "leo");

string outMessage = string.Empty;

bool issuccess = addressBookMail.UnSubscribe(out outMessage);

if (!issuccess)

{

Console.WriteLine(outMessage);

}

**AddressBookMessage 类**

**初始化类**

**IAppConfig appConfig = new MessageConfig("AppId", "AppKey");**

**AddressBookMessage addressbook = new AddressBookMessage(config);**

**使用指引**

**方法 描述**

**SetAddress 设置联系人11位手机号码**

**SetAddressbook 设置目标地址薄标识**

**Subscribe 订阅或添加联系人**

**UnSubscribe 退订或移除联系人**

**GetSender 获取发送方式**

**代码示例**

使用 ADDRESSBOOKMessage 类 发送 addressbook/message/subscribe 请求，添加订阅联系人或向目标地址薄中添加联系人

IAppConfig appConfig = new MessageConfig("AppId", "AppKey");

AddressBookMessage addressBookMessage = new AddressBookMessage(appConfig);

string returnMessage = string.Empty;

addressBookMessage.SetAddress("your phone number");

if (addressBookMessage.Subscribe(out returnMessage) == false)

{

Console.WriteLine(returnMessage);

}

使用 ADDRESSBOOKMessage 类 发送 addressbook/message/unsubscribe 请求，添加退订联系人或从目标地址薄中移除联系人

IAppConfig appConfig = new MessageConfig("AppId", "AppKey");

AddressBookMessage addressBookMessage = new AddressBookMessage(appConfig);

string returnMessage = string.Empty;

addressBookMessage.SetAddress("your phone number");

if (addressBookMessage.UnSubscribe(out returnMessage) == false)

{

Console.WriteLine(returnMessage);

}

**MessageMultiXSend 类**

**初始化类**

**IAppConfig appConfig = new MessageConfig("AppId", "AppKey");**

**MessageMultiXSend messageMultiSend = new MessageMultiXSend(appConfig);**

**使用指引**

**方法 描述**

**GetSender 获取发送方式**

**SetProject 设置短信项目标记**

**SetMulti 设置对个发送对象信息**

**MultiXSend 同时向多个联系人发送短信**

**代码示例**

使用 MessageMultiXSend同时向多个联系人发送短信

IAppConfig appConfig = new MessageConfig("AppId", "AppKey");

MessageMultiXSend messageMultiSend = new MessageMultiXSend(appConfig);

messageMultiSend.SetProject("MApf82");

Dictionary<string, string> vars = new Dictionary<string, string>();

vars.Add("code", "你好");

messageMultiSend.SetMulti(new List<MultiMessageItem>() {

new MultiMessageItem() { to="one phone number"},

new MultiMessageItem() { to="another phone number", vars = vars}

});

string returnMessage = string.Empty;

if (messageMultiSend.MultiXSend(out returnMessage) == false)

{

Console.WriteLine(returnMessage);

}

**VoiceSend 类**

**初始化类**

**IAppConfig appConfig = new VoiceAppConfig("APP\_ID","APP\_Key");**

**VoiceVerify voiceVerify = new VoiceVerify(appConfig);**

**使用指引**

**方法 描述**

**AddTo 添加11位手机号码**

**SetCode 添加语音播报的4位数字验证码**

**GetSender 获取发送方式**

**Verify 发送语音验证码，进行语言验证**

**代码示例**

IAppConfig appConfig = new VoiceAppConfig("APP\_ID","APP\_Key");

**VoiceVerify** **voiceVerify** = new **VoiceVerify** (appConfig);

**voiceVerify**.AddTo("XXXXXXXXXXX");

// 此数字为语音播报的4位数字验证码

**voiceVerify**.AddCode("1234");

string returnMessage = string.Empty;

if (**voiceVerify**.Verify(out returnMessage) == false)

{

Console.WriteLine(returnMessage);

}