

TBaaS

SDK 文档

产品文档



腾讯云

【 版权声明 】

©2013-2021 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

SDK 文档

Hyperledger Fabric

更新历史

Go SDK

Java SDK

Python SDK

SDK 文档

Hyperledger Fabric

更新历史

最近更新时间：2020-05-13 11:09:51

发布时间	更新说明	版本说明
2020-02-05	第四次发布	更新 SDK 文档代码用例
2019-08-14	第三次发布	添加 Python SDK 代码注释
2019-02-15	第二次发布	-
2018-12-03	第一次发布	-

Go SDK

最近更新时间：2020-02-05 12:25:13

简介

欢迎使用腾讯云 TBaaS 产品开发者工具套件（SDK）3.0，SDK3.0 是云 API3.0 平台的配套工具。为方便 Go 开发者调试和接入腾讯云 TBaaS 产品 API，这里向您介绍适用于 Go 的腾讯云 TBaaS 产品开发工具包，并提供首次使用开发工具包的简单示例。让您快速获取腾讯云 TBaaS 产品 Go SDK 并开始调用。

依赖环境

1. 依赖环境：Go 1.9版本及以上，并设置好 GOPATH 等必须的环境变量。
2. 通过腾讯云控制台开通 TBaaS 产品。
3. 获取 [SecretID](#)、[SecretKey](#) 以及调用地址（tbaas.tencentcloudapi.com）。

获取安装

安装 Go SDK 和第一次使用云 API 之前，用户需要在腾讯云控制台上申请并获取安全凭证。安全凭证包括 SecretID 和 SecretKey。SecretID 用于标识 API 调用者的身份，SecretKey 用于加密签名字符串和服务端验证签名字符串的密钥。SecretKey 必须严格保管，避免泄露。

通过 go get 安装（推荐）

使用语言自带的工具安装 SDK：

```
go get -u github.com/tencentcloud/tencentcloud-sdk-go
```

通过源码包安装

1. 前往 [Github 代码托管地址](#) 下载最新代码。
2. 将获取到的源码包解压缩，并安装到 `$GOPATH/src/github.com/tencentcloud` 目录下。

接口列表

接口名称	接口功能
Invoke	新增交易（支持同步模式和异步模式）
Query	查询交易

接口名称	接口功能
GetInvokeTx	查询 Invoke 异步调用结果
GetBlockList	查询区块列表
GetBlockTransactionListForUser	获取区块内的交易列表
GetClusterSummary	获取区块链网络概要
GetLatesdTransactionList	获取最新交易列表
GetTransactionDetailForUser	获取交易详情
ApplyUserCert	申请用户证书
DownloadUserCert	下载用户证书
SrvInvoke	trustsql 服务统一接口
BlockByNumberHandler	按块高查询区块信息
DeployDynamicContractHandler	动态部署合约
GetBlockListHandler	查询区块列表
GetTransByHashHandler	根据交易哈希查询交易信息
GetTransListHandler	查询交易列表
SendTransactionHandler	发送交易
TransByDynamicContractHandler	根据动态部署的合约发送交易

示例

以新增交易（Invoke）接口为例：

```
package main
import (
    "fmt"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/errors"
    "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/common/profile"
    tbaas "github.com/tencentcloud/tencentcloud-sdk-go/tencentcloud/tbaas/v20180416"
```

```
)

func main() {
// 必要步骤：
// 实例化一个认证对象，入参需要传入腾讯云账户密钥对secretId, secretKey。
credential := common.NewCredential("secretId", "secretKey")
// 实例化一个客户端配置对象，可以指定超时时间等配置
cpf := profile.NewClientProfile()
// SDK有默认的超时时间，非必要请不要进行调整。
// 如有需要请在代码中查阅以获取最新的默认值。
cpf.HttpProfile.ReqTimeout = 10
// 设置访问域名
// SDK会自动指定域名。通常是不需要特地指定域名的，但是如果您访问的是金融区的服务，
// 则必须手动指定域名，例如云服务器上海金融区域名：tbaas.ap-shanghai-fsi.tencentcloudap
i.com
cpf.HttpProfile.Endpoint = "tbaas.tencentcloudapi.com"
// 实例化Tbaas的client对象
// 第二个参数是地域信息，根据资源所属地域填写相应的地域信息，例如广州地域的资源可以直接填写字符串ap-guangzhou，或者引用预设的常量
client, _ := tbaas.NewClient(credential, regions.Guangzhou, cpf)
// 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
request := tbaas.NewInvokeRequest()
params := "{\"Module\": \"transaction\", \"Operation\": \"invoke\", \"ClusterId\": \"251005746ctestenv\", \"ChaincodeName\": \"pettycc1\", \"ChannelName\": \"pettyc1\", \"Peers\": [{\"PeerName\": \"peer0.pettycorg.ctestenv\", \"OrgName\": \"pettyc0rg\"}], \"FuncName\": \"invoke\", \"Args\": [\"a\", \"b\", \"10\"], \"AsyncFlag\": 0, \"GroupName\": \"pettyc0rg\"}"
err := request.FromJsonString(params)
if err != nil {
panic(err)
}
// 通过client对象调用想要访问的接口，需要传入请求对象
response, err := client.Invoke(request)
// 处理异常
if _, ok := err.(*errors.TencentCloudSDKError); ok {
fmt.Printf("An API error has returned: %s", err)
return
}
}
```

```
// 非SDK异常，直接失败。实际代码中可以加入其他的处理。  
if err != nil {  
    panic(err)  
}  
  
// 打印返回的json字符串  
fmt.Printf("%s", response.ToJsonString())  
}
```


Java SDK

最近更新时间：2021-01-29 09:15:36

简介

欢迎使用腾讯云 TBaaS 产品开发者工具套件（SDK）3.0，SDK3.0 是云 API3.0 平台的配套工具。为方便 Java 开发者调试和接入腾讯云 TBaaS 产品 API，这里向您介绍适用于 Java 的腾讯云 TBaaS 产品开发工具包，并提供首次使用开发工具包的简单示例。让您快速获取腾讯云 TBaaS 产品 Java SDK 并开始调用。

依赖环境

1. 依赖环境：JDK 7版本及以上。
2. 通过腾讯云控制台开通 TBaaS 产品。
3. 获取 [SecretID](#)、[SecretKey](#) 以及调用地址（tbaas.tencentcloudapi.com）。

获取安装

安装 Java SDK 和第一次使用云 API 之前，用户需要在腾讯云控制台上申请并获取安全凭证。安全凭证包括 SecretID 和 SecretKey。SecretID 用于标识 API 调用者的身份，SecretKey 用于加密签名字符串和服务端验证签名字符串的密钥。SecretKey 必须严格保管，避免泄露。

通过 Maven 安装（推荐）

Java SDK 推荐通过 Maven 安装。Maven 是 Java 的依赖管理工具，支持您项目所需的依赖项，并将其安装到项目中。关于 Maven 详细介绍可参考 [Maven 官网](#)。

1. 前往 [Maven 官网](#) 下载对应系统的 Maven 安装包，并进行安装。
2. 在 Maven pom.xml 添加以下内容，为您的项目添加 Maven 依赖项。

```
<dependency>
<groupId>com.tencentcloudapi</groupId>
<artifactId>tencentcloud-sdk-java</artifactId>
<version>3.0.1</version>
</dependency>
```

🔍 说明：

<version> 标签中的版本号为参考示例，请在 Maven 仓库上找到最新的版本进行填写。

3. 引用方法可参考 [示例](#)。

通过源码包安装

1. 前往 [Github 代码托管地址](#) 下载源码压缩包。
2. 将获取到的源码包解压缩到您项目合适的位置。
3. 将 vendor 目录下的 jar 包拷贝到 Java 可找到的路径中。
4. 引用方法可参考 [示例](#)。

接口列表

接口名称	接口功能
Invoke	新增交易（支持同步模式和异步模式）
Query	查询交易
GetInvokeTx	查询 Invoke 异步调用结果
GetBlockList	查询区块列表
GetBlockTransactionListForUser	获取区块内的交易列表
GetClusterSummary	获取区块链网络概要
GetLatesdTransactionList	获取最新交易列表
GetTransactionDetailForUser	获取交易详情
ApplyUserCert	申请用户证书
DownloadUserCert	下载用户证书
SrvInvoke	trustsql 服务统一接口
BlockByNumberHandler	按块高查询区块信息
DeployDynamicContractHandler	动态部署合约
GetBlockListHandler	查询区块列表
GetTransByHashHandler	根据交易哈希查询交易信息
GetTransListHandler	查询交易列表
SendTransactionHandler	发送交易

接口名称	接口功能
TransByDynamicContractHandler	根据动态部署的合约发送交易

示例

以新增交易（Invoke）接口为例：

```
import com.tencentcloudapi.common.Credential;
import com.tencentcloudapi.common.profile.ClientProfile;
import com.tencentcloudapi.common.profile.HttpProfile;
import com.tencentcloudapi.common.exception.TencentCloudSDKException;
// 导入对应产品模块的client
import com.tencentcloudapi.tbaas.v20180416.TbaasClient;
// 导入要请求接口对应的request response类
import com.tencentcloudapi.tbaas.v20180416.models.InvokeRequest;
import com.tencentcloudapi.tbaas.v20180416.models.InvokeResponse;
public class InvokeTransaction
{
public static void main(String [] args) {
try{
// 实例化一个认证对象，入参需要传入腾讯云账户密钥对secretId, secretKey
Credential cred = new Credential("secretId", "secretKey");
// 设置访问域名
// SDK会自动指定域名。通常是不需要特地指定域名的，但是如果您访问的是金融区的服务，
// 则必须手动指定域名，例如云服务器的上海金融区域名： tbaas.ap-shanghai-fsi.tencentcloudapi.com
HttpProfile httpProfile = new HttpProfile();
httpProfile.setEndpoint("tbaas.tencentcloudapi.com");
// 实例化Tbaas的client对象
ClientProfile clientProfile = new ClientProfile();
clientProfile.setHttpProfile(httpProfile);
// 第二个参数是地域信息，根据资源所属地域填写相应的地域信息，例如广州地域的资源可以直接填写字符串ap-guangzhou，或者引用预设的常量
TbaasClient client = new TbaasClient(cred, "ap-guangzhou", clientProfile);
// 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
String params = "{\"Module\":\"transaction\",\"Operation\":\"invoke\",\"ClusterId\": \"251005746ctestenv\", \"ChaincodeName\":\"pettycc1\", \"ChannelName\":\"pettyc1
```

```
\", \"Peers\": [{\"PeerName\": \"peer0.pettycorg.ctestenv\", \"OrgName\": \"pettycOrg\"}], \"FuncName\": \"invoke\", \"Args\": [\"a\", \"b\", \"10\"], \"AsyncFlag\": 0, \"GroupName\": \"pettycOrg\"}";  
InvokeRequest req = InvokeRequest.fromJsonString(params, InvokeRequest.class);  
// 通过client对象调用想要访问的接口, 需要传入请求对象  
InvokeResponse resp = client.Invoke(req);  
// 输出json格式的字符串回包  
System.out.println(InvokeResponse.toJsonString(resp));  
} catch (TencentCloudSDKException e) {  
System.out.println(e.toString());  
}  
}  
}
```

Python SDK

最近更新时间：2020-02-05 12:26:29

简介

欢迎使用腾讯云 TBaaS 产品开发者工具套件（SDK）3.0，SDK3.0 是云 API3.0 平台的配套工具。为方便 Python 开发者调试和接入腾讯云 TBaaS 产品 API，这里向您介绍适用于 Python 的腾讯云 TBaaS 产品开发工具包，并提供首次使用开发工具包的简单示例。让您快速获取腾讯云 TBaaS 产品 Python SDK 并开始调用。

依赖环境

1. 依赖环境：Python 2.7到3.6版本。
2. 通过腾讯云控制台开通 TBaaS 产品。
3. 获取 [SecretID](#)、[SecretKey](#) 以及调用地址（tbaas.tencentcloudapi.com）。

获取安装

安装 Python SDK 和第一次使用云 API 之前，用户需要在腾讯云控制台上申请并获取安全凭证。安全凭证包括 SecretID 和 SecretKey。SecretID 用于标识 API 调用者的身份，SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。SecretKey 必须严格保管，避免泄露。

通过 Pip 安装（推荐）

您可以通过执行以下命令，将腾讯云 API Python SDK 安装到您的项目中。如果您的项目环境尚未安装 pip，请参见 [pip官网](#) 进行安装。

```
pip install tencentcloud-sdk-python
```

⚠ 注意：

如果您的项目环境中同时搭建 Python2 和 Python3 环境，在 Python3 环境下，请使用 pip3 命令进行安装。

通过源码包安装

1. 前往 Github 代码托管地址 下载最新代码。
2. 将获取到的源码包解压缩，并执行以下命令进行安装。

```
$ cd tencentcloud-sdk-python
$ python setup.py install
```

接口列表

接口名称	接口功能
Invoke	新增交易（支持同步模式和异步模式）
Query	查询交易
GetInvokeTx	查询 Invoke 异步调用结果
GetBlockList	查询区块列表
GetBlockTransactionListForUser	获取区块内的交易列表
GetClusterSummary	获取区块链网络概要
GetLatesdTransactionList	获取最新交易列表
GetTransactionDetailForUser	获取交易详情
ApplyUserCert	申请用户证书
DownloadUserCert	下载用户证书
SrvInvoke	trustsql 服务统一接口
BlockByNumberHandler	按块高查询区块信息
DeployDynamicContractHandler	动态部署合约
GetBlockListHandler	查询区块列表
GetTransByHashHandler	根据交易哈希查询交易信息
GetTransListHandler	查询交易列表
SendTransactionHandler	发送交易
TransByDynamicContractHandler	根据动态部署的合约发送交易

示例

以新增交易（Invoke）接口为例：

```
# -*- coding: utf-8 -*-
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException

# 导入对应产品模块的client models。
from tencentcloud.tbaas.v20180416 import tbaas_client, models

try:
# 实例化一个认证对象，入参需要传入腾讯云账户密钥对secretId, secretKey
cred = credential.Credential("secretId", "secretKey")

# 配置访问域名
# SDK会自动指定域名。通常是不需要特地指定域名的，但是如果您访问的是金融区的服务，
# 则必须手动指定域名，例如云服务器的上海金融区域名： tbaas.ap-shanghai-fsi.tencentcloudap
i.com
httpProfile = HttpProfile()
httpProfile.endpoint = "tbaas.tencentcloudapi.com"

# 实例化Tbaas的client对象
clientProfile = ClientProfile()
clientProfile.httpProfile = httpProfile
#第二个参数是地域信息，根据资源所属地域填写相应的地域信息，例如广州地域的资源可以直接填写字符串
ap-guangzhou，或者引用预设的常量
client = tbaas_client.TbaasClient(cred, "ap-guangzhou", clientProfile)

# 实例化一个请求对象，根据调用的接口和实际情况，可以进一步设置请求参数
params = '{"Module":"transaction","Operation":"invoke","ClusterId":"251005746ctestenv","ChaincodeName":"pettycc1","ChannelName":"pettyc1","Peers":[{"PeerName":"peer0.pettyccorg.ctestenv","OrgName":"pettyc0rg"}],"FuncName":"invoke","Args":["a","b","10"],"AsyncFlag":0,"GroupName":"pettyc0rg"}'
req = models.InvokeRequest()
# 调用InvokeRequest的from_json_string方法，使用params初始化req对象
req.from_json_string(params)

# 通过client对象调用想要访问的接口，需要传入请求对象
resp = client.Invoke(req)
```

```
# 输出json格式的字符串回包
print(resp.to_json_string())

except TencentCloudSDKException as err:
    print(err)
```