

目录

目录	1
介绍	2
分布式事务服务KDTX OpenAPI文档	2
服务信息	2
服务信息	2
Region及可用区	2
核心概念	2
计费类型	2
按日用量实时付费	2
公共参数	2
签名机制	2
Python Demo	2
Java Demo	3
Go Demo	5
错误码	6
错误码	6
OK	6
INVALID_ACTION	6
INVALID_PARAMETER_COMBINATION	6
INVALID_PARAMETER_VALUE	6
MISSING_PARAMETER	6
NOT_FOUND	6
MALFORMED_QUERY_STRING	6
INTERNAL_FAILURE	6
SERVICE_UNAVAILABLE	6
BACKING_UP	6
INSTANCE_TASK	7
OPERATION_DENIED_CREATE_BACKUP	7
INVALID_PACKAGES	7
查看监控数据	7
查看事务分组列表	7
查询分支事务	8
查询全局事务	8
查询指定用户全局事务概览信息（概览页）	9
新建事务分组	9

介绍

分布式事务服务KDTX OpenAPI文档

本文档详细介绍了分布式事务服务KDTX 的OpenAPI的调用方式和各种相关接口。

服务信息

服务信息

- 服务地址
<http://kdtx.api.ksyun.com>
- 通讯协议
支持HTTP或HTTPS协议
- 请求方法
详见具体API接口
- 请求参数
每个请求都需要包含公共的请求参数和操作所特有的请求参数，详见具体API接口
- 编码方式
请求及结果返回都使用UTF-8字符集编码
- 返回结果
仅支持json返回结果

Region及可用区

各个region都通过同一个endpoint来访问API服务：krds.api.ksyun.com

Region名称	类型	Region代码	可用区
华东1（上海）	VPC机房	cn-shanghai-2	cn-hangzhou-2b

核心概念

计费类型

金山云分布式事务支持按日用量实时付费计费方式。在API接口中可以通过设置BillType参数来指定计费类型。

按日用量实时付费

采取按量付费的计费方式。实际事务总数等于全局事务数和分支事务数的总和。按量付费是先使用，后付费。每天按KDTX实际事务数计费。

参数名称	中文名称	类型	备注
BillType	计费方式	String	默认值：86，取值范围：86（按日计费）。

公共参数

支持GET和POST两种HTTP方法。

固定请求域名：`{service}.{Region}.api.ksyun.com`（不指定区域则默认为cn-beijing-6，若要指定区域：`{service}.{Region}.api.ksyun.com`）

详情请参考：[公共参数](#)

签名机制

数据库的openAPI支持GET和POST两种HTTP方法，具体流程请参见[签名机制](#)。

Python Demo

```
import hashlib
import hmac
```

```

import time
import urllib.request

import requests

# Python 3.9.6
def http_execute(methed, host, heads, data=None):
    url = 'https://%s/?' % host
    for key in data:
        url += urllib.request.quote(key, '~') + '=' + urllib.request.quote(str(data[key]), '~') + '&'
    response = requests.request(methed, url, headers=heads, data=None)
    return response;

# sigon方法
def sign(params, secret_key):
    str_encode = ''
    param_keys = sorted(params.keys())
    for key in param_keys:
        str_encode += urllib.request.quote(key, '~') + '=' + urllib.request.quote(str(params[key]), '~') + '&'
    return hmac.new(bytes(secret_key, 'utf-8'), bytes(str_encode[:-1], 'utf-8'), hashlib.sha256).hexdigest()

class ksyun_iam_api_tools():
    """金山云API相关"""

    def __init__(self):
        self.AK = 'your_ak'
        self.SK = 'your_sk'
        self.SERVICE = "krds"
        self.REGION = "cn-beijing-6"
        self.HOST = "%s.%s.api.ksyun.com" % (self.SERVICE, self.REGION)
        self.additional_headers = {'Accept': 'application/json'}

    def describe_db_instances(self):
        data = {
            'Accesskey': self.AK,
            'Service': 'krds',
            'Action': 'DescribeDBInstances',
            'Version': '2016-07-01',
            'Timestamp': time.strftime("%Y-%m-%dT%H:%M:%SZ", time.gmtime()), # 使用UTC格式的时间
            'SignatureVersion': '1.0',
            'SignatureMethod': 'HMAC-SHA256',
        }
        data["Signature"] = sign(data, self.SK)
        return http_execute("GET", self.HOST, self.additional_headers, data)

if __name__ == '__main__':
    api = ksyun_iam_api_tools()
    instances = api.describe_db_instances()
    print(instances.text)

```

Java Demo

```

import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;
import org.apache.commons.codec.digest.HmacAlgorithms;
import org.apache.commons.codec.digest.HmacUtils;

import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

// jdk1.8.0_241
public class ram_demo {
    private static final Pattern ENCODED_CHARACTERS_PATTERN;

    static {
        StringBuilder pattern = new StringBuilder();
        pattern.append(Pattern.quote("+"))
            .append("|")
            .append(Pattern.quote("*"))
            .append("|")
            .append(Pattern.quote("%7E"))
            .append("|")
            .append(Pattern.quote("%2F"));
        ENCODED_CHARACTERS_PATTERN = Pattern.compile(pattern.toString());
    }
}

```

```

/**
 * 注意空格( )会编码成+号, 所以+最后需要替换成%20 (%20为空格)
 * 在URLEncode后需对三种字符替换: 加号(+) 替换成 %20、星号(*) 替换成 %2A、 %7E 替换回波浪号(~)
 */
private static String urlEncode(final String value, final boolean path) {
    if (value == null) {
        return "";
    }

    try {

        String encoded = URLEncoder.encode(value, StandardCharsets.UTF_8.name());
        Matcher matcher = ENCODED_CHARACTERS_PATTERN.matcher(encoded);
        StringBuffer buffer = new StringBuffer(encoded.length());

        while (matcher.find()) {
            String replacement = matcher.group(0);
            if ("+".equals(replacement)) {
                replacement = "%20";
            } else if ("*".equals(replacement)) {
                replacement = "%2A";
            } else if ("%7E".equals(replacement)) {
                replacement = "~";
            } else if (path && "%2F".equals(replacement)) {
                replacement = "/";
            }

            matcher.appendReplacement(buffer, replacement);
        }
        matcher.appendTail(buffer);
        return buffer.toString();
    } catch (UnsupportedEncodingException ex) {
        throw new RuntimeException(ex);
    }
}

private static String getCanonicalizedQueryString(Map<String, Object> params) {
    final Map<String, String> tmap = new TreeMap<>();
    for (Map.Entry<String, Object> entry : params.entrySet()) {
        String key = entry.getKey();
        Object value = entry.getValue();
        String encodedParamName = urlEncode(key, false);
        String encodedValues = urlEncode(value.toString(), false);
        tmap.put(encodedParamName, encodedValues);
    }

    final StringBuilder sb = new StringBuilder();
    for (Map.Entry<String, String> entry : tmap.entrySet()) {
        if (sb.length() > 0) {
            sb.append("&");
        }
        sb.append(entry.getKey()).append('=').append(entry.getValue());
    }
    return sb.toString();
}

public static String signature(Map<String, Object> params, String secretKey) {
    String canonicalizedQueryString = getCanonicalizedQueryString(params);
    return new HmacUtils(HmacAlgorithms.HMAC_SHA_256, secretKey).hmacHex(canonicalizedQueryString);
}

public static String utc2Local(String utcTime, String utcTimePatten, String localTimePatten) throws ParseException {
    SimpleDateFormat utcFormatter = new SimpleDateFormat(utcTimePatten);
    utcFormatter.setTimeZone(TimeZone.getTimeZone("UTC")); //时区定义并进行时间获取
    Date gpsUTCDate = null;
    gpsUTCDate = utcFormatter.parse(utcTime);
    SimpleDateFormat localFormatter = new SimpleDateFormat(localTimePatten);
    localFormatter.setTimeZone(TimeZone.getDefault());
    String localTime = localFormatter.format(gpsUTCDate.getTime());
    return localTime;
}

public static void test1() throws IOException {
    String accesskey = "your_ak";
    String secretKey = "your_sk";

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'");
    TimeZone zone = TimeZone.getTimeZone("UTC");
    Calendar cal = Calendar.getInstance(zone);
    sdf.setTimeZone(zone);

    Map<String, Object> params = new HashMap<>();
    params.put("Accesskey", accesskey); //公共参数

```

```

params.put("Service", "krds"); //公共参数
params.put("Action", "DescribeDBInstances"); //公共参数
params.put("Version", "2016-07-01"); //公共参数
params.put("Timestamp", sdf.format(cal.getTime())); //公共参数
params.put("SignatureVersion", "1.0"); //公共参数
params.put("SignatureMethod", "HMAC-SHA256"); //公共参数
params.put("Signature", signature(params, secretKey));

String product_name = "krds";
String region = "cn-beijing-6";

String url = "https://" + product_name + "." + region + ".api.ksyun.com/?" + getCanonicalizedQueryString(params);
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
Request request = new Request.Builder()
    .url(url)
    .method("GET", null)
    .addHeader("Accept", "application/json")
    .build();
Response response = client.newCall(request).execute();

System.out.println(new String(response.body().bytes()));
}

public static void main(String[] args) {
    try {
        test1();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}
}

```

Go Demo

```

package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "fmt"
    "io/ioutil"
    "net/http"
    "net/url"
    "strings"
    "time"
)

// go version go1.16.5 darwin/amd64

//获取signature签名
func sign(params url.Values, sk string) string {
    strEncode := params.Encode()
    strEncode = strings.Replace(strEncode, "+", "%20", -1)
    h := hmac.New(sha256.New, []byte(sk))
    h.Write([]byte(strEncode))
    return hex.EncodeToString(h.Sum(nil))
}

func main() {

    params := url.Values{}
    params.Add("Accesskey", "your_ak")
    params.Add("Service", "krds")
    params.Add("Action", "DescribeDBInstances")
    params.Add("Version", "2016-07-01")
    params.Add("Timestamp", time.Now().UTC().Format("2006-01-02T15:04:05Z")) //通过time.Now().UTC().Format("2006-01-02T15:04:05Z")实现
    params.Add("SignatureVersion", "1.0")
    params.Add("SignatureMethod", "HMAC-SHA256")
    sk := "your_sk"
    signature := sign(params, sk)
    params.Add("Signature", signature)

    strEncode := params.Encode()
    strEncode = strings.Replace(strEncode, "+", "%20", -1)

    url := "https://krds.cn-beijing-6.api.ksyun.com/?" + strEncode
    method := "GET"
    client := &http.Client{}
    req, _ := http.NewRequest(method, url, nil)
    req.Header.Add("Accept", "application/json")
}

```

```
res, _ := client.Do(req)
defer res.Body.Close()
body, _ := ioutil.ReadAll(res.Body)
fmt.Println(string(body))
}
```

错误码

错误码

OK

成功

- HttpStatusCode: 200

INVALID_ACTION

操作无效, 请检查操作是否符合规范

- HttpStatusCode: 400

INVALID_PARAMETER_COMBINATION

参数绑定错误

- HttpStatusCode: 400

INVALID_PARAMETER_VALUE

参数值必要或者参数值格式不正确

- HttpStatusCode: 400

MISSING_PARAMETER

参数必填

- HttpStatusCode: 400

NOT_FOUND

资源不存在, 或者已被删除

- HttpStatusCode: 404

MALFORMED_QUERY_STRING

请求包含语法错误

- HttpStatusCode: 404

INTERNAL_FAILURE

服务器内部错误, 联系技术支持解决

- HttpStatusCode: 500

SERVICE_UNAVAILABLE

服务暂不可用, 请稍后重试

- HttpStatusCode: 503

BACKING_UP

实例备份中, 无法进行其他操作, 请稍后重试

- HttpStatusCode: 403

INSTANCE_TASK

实例正在任务中，任务中包括重启中、主从切换中等

- HttpStatusCode: 409

OPERATION_DENIED_CREATE_BACKUP

一个实例至多支持5个手动备份

- HttpStatusCode: 400

INVALID_PACKAGES

不支持购买非标套餐配置，请联系客服人员

- HttpStatusCode: 400

查看监控数据

接口	method	url			
GET	/kdtx				
输入项中文名称	字段名	数据类型	输入属性	取值范围及备注	
Action	Action	String	必输	InspectDistributeTransactionGroupsMonitor	
查询开始时间	StartTime	long			
查询结束时间	EndTime	long			
GroupId	long				
header中文名	字段名	输入属性	取值范围		
区域	X-KSC-REGION				
账户名称	X-KSC-ACCOUNT-ID				
输出项中文名称	字段名	数据类型	输出属性	取值范围及备注	
msg	msg	String			
status	status	int			
data	data	对象		timestamp: 时间戳; globalTransCount: 时间戳对应的全局事务数; branchTransCount: 时间戳对应的分支事务数; [{ "timestamp": 1561608838000, "globalTxCount": 10, "branchTxCount": 30 }, { "timestamp": 1561608838000, "globalTxCount": 10, "branchTxCount": 30 }]	

查看事务分组列表

接口	method	url			
GET	/kdtx				
输入项中文名称	字段名	数据类型	输入属性	取值范围及备注	
Action	Action	String	必输	InspectDistributeTransactionGroups	
请求页码	Page			默认为第一页	
每页展示数量	Size			默认每页10条记录	
header中文名	字段名	输入属性	取值范围		
账户名称	X-KSC-ACCOUNT-ID				
区域	X-KSC-REGION				
输出项中文名称	字段名	数据类型	输出属性	取值范围及备注	
msg	msg	String			
status	status	int			
page	page	int			
size	size	int			

totalPage totalPage int
totalCount totalCount int
t t

data data 对象

groupId: 记录ID, 唯一标识一条记录; region: region名称; groupName: 事务分组名称; globalTransCount: 全局事务总数; branchTransCount: 分支事务总数; createTime: 事务分组创建时间; [{ "groupId": 1, "region": "cn-beijing-1", "groupName": "order-kdtx-service-group", "groupCount": 1, "globalTransCount": 20, "branchTransCount": 30, "createTime": 1561608838000, "token": "dfaslwerwekj334234jlljadf" }, { "groupId": 2, "region": "cn-beijing-1", "groupName": "order-kdtx-service-group", "groupCount": 1, "globalTransCount": 20, "branchTransCount": 30, "createTime": 1561608838000, "token": "dfaslwerwekj334234jlljadf" }]

查询分支事务

接口 method url
GET /kdtx

输入项中文名称	字段名	数据类型	输入属性	取值范围及备注
Action	Action	String	必输	InspectBranchDistributeTransactions
请求页码	Page			默认为第一页
每页展示数量	Size			默认每页10条记录
事务状态	State	string	选填	running/commit/rollback/error
全局事务ID	Xid	string	选填	
分支事务ID	BranchId	string	选填	
事务分组ID	GroupId	long		

header中文名称	字段名	输入属性	取值范围
账户名称	X-KSC-ACCOUNT-ID		
区域	X-KSC-REGION		

输出项中文名称	字段名	数据类型	输出属性	取值范围及备注
msg	msg	String		
status	status	int		
page	page	int		
size	size	int		
totalPage	totalPage	int		
totalCount	totalCount	int		
t	t			

data data 对象

id: 记录ID, 唯一标识一条记录; xid: 全局事务ID; branchId: 分支事务ID; state: 分支事务状态: commit/rollback/running/error; stateText: 分支服务状态展示文本: 已提交、已回滚、运行中、异常; lockKey: 全局锁标识; resourceId: 资源ID; clientId: 客户端ID; branchType: 分支事务类型: 0自动事务, 1手动事务; branchTypeText: 分支事务类型展示文本; msg: 如果为异常状态, 则记录异常信息; region: 地域标识; [{ "id": 1, "xid": "23429-afas-2134-eqwrq", "branchId": "23429-afas-2134-eqwrq", "state": "error", "stateText": "异常", "lockKey": "poc_account:100", "resourceId": "jdbc:mysql://product:112.23.12.23:8992", "clientId": "product:112.23.12.23:8992", "branchType": 0, "branchTypeText": "AT", "msg": "rollback timeout", "region": "cn-beijing-1" }, { "id": 2, "xid": "23429-afas-2134-eqwrq", "branchId": "23429-afas-2134-eqwrq", "state": "commit", "stateText": "已提交", "lockKey": "poc_account:100", "resourceId": "jdbc:mysql://product:112.23.12.23:8992", "clientId": "product:112.23.12.23:8992", "branchType": 1, "branchTypeText": "MT", "msg": "success", "region": "cn-beijing-1" }]

查询全局事务

接口 method url
GET /kdtx

输入项中文名称	字段名	数据类型	输入属性	取值范围及备注
Action	Action	String	必输	InspectGlobalDistributeTransactions

请求页码	Page			默认为第一页
每页展示数量	Size			默认每页10条记录
事务状态	State	string	选填	running/commit/rollback/error
全局事务ID	Xid	string	选填	
全局事务名称	Name	string	选填	
应用ID	AppId	string		
groupId	GroupId	long		

header中文名 字段名 输入属性 取值范围

账户名称	X-KSC-ACCOUNT-ID
区域	X-KSC-REGION

输出项中文名称 字段名 数据类型 输出属性 取值范围及备注

msg	msg	String		
status	status	int		
page	page	int		
size	size	int		
totalPage	totalPage	int		
totalCount	totalCount	int		

data	data	对象		
------	------	----	--	--

xid: 全局事务ID; name: 全局事务名称; state: 全局事务状态; stateText: 分支服务状态展示文本; 已提交、已回滚、运行中、异常; commit/rollback/running/error; appId: 应用ID; startTime: 事务开始时间; timeout: 事务超时时间; msg: 如果为异常状态, 则记录异常信息;

```
[ { "id": 1, "xid": "23429-adas-2134-eqwrq", "name": "payTheBill (com.ksyun.demo.order.controller.vo.PayReq)", "state": "error", "stateText": "异常", "appId": "order", "startTime": 1561608838000, "timeout": 6000, "msg": "rollback timeout", "region": "cn-beijing-1" }, { "id": 2, "xid": "23429-adas-2134-eqwrq", "name": "payTheBill (com.ksyun.demo.order.controller.vo.PayReq)", "state": "committed", "stateText": "已提交", "appId": "order", "startTime": 1561608838000, "timeout": 6000, "msg": "success", "region": "cn-beijing-1" } ]
```

查询指定用户全局事务概览信息（概览页）

接口 method url
GET /kdtx

输入项中文名称 字段名 数据类型 输入属性 取值范围及备注
动作名称 Action String 必输 Resume

header中文名 字段名 输入属性 取值范围

账户名称	X-KSC-ACCOUNT-ID
------	------------------

输出项中文名称 字段名 数据类型 输出属性 取值范围及备注

status	status	int		
msg	msg	String		

data	data	对象		
------	------	----	--	--

groupCount: 事务分组总数; globalTransCount: 全局事务总数; branchTransCount: 分支事务总数; regions: 分region统计信息; region: region名称; groupCount: 事务分组数; globalTrans: 全局事务统计; branchTrans: 分支事务统计; total: 事务总数; committed: 已提交事务数; rollbacked: 已回滚事务数; running: 运行中事务数; error: 异常事务数;

```
{ "groupCount": 2, "globalTxCount": 10, "branchTxCount": 30, "regions": [ { "region": "华北1(北京)", "groupCount": 1, "globalTx": { "total": 5, "committed": 3, "rollbacked": 1, "running": 1, "error": 0 }, "branchTx": { "total": 5, "committed": 3, "rollbacked": 1, "running": 1, "error": 0 } } ] }
```

新建事务分组

接口 method url
GET /kdtx

输入项中文名称 字段名 数据类型 输入属性 取值范围及备注

Action Action String 必输 CreateDistributeTransactionGroup

分组名称 GroupName

计费类型 BillType

ProductWhat 1. 正式 2. 试用

header中文名 字段名 输入属性 取值范围

账户名称 X-KSC-ACCOUNT-ID

区域 X-KSC-REGION

输出项中文名称 字段名 数据类型 输出属性 取值范围及备注

msg msg String

status status int

data data 对象
 groupId: 分组ID; group
 name: 分组名称; regio
 n: 名称; ch
 argeType:
 计费类型,
 预留字段,
 给1即可;
 { "groupCount": 2, "globalTxCount": 10, "branchTxCount": 30, "reg
 ions": [{ "region": "华北1(北京)", "groupCount": 1, "globalTx"
 : { "total": 5, "committed": 3, "rollbacked": 1, "running": 1, "e
 rror": 0 }, "branchTx": { "total": 5, "committed": 3, "rollbacke
 d": 1, "running": 1, "error": 0 } }, { "region": "北京5区", "group
 Count": 1, "globalTx": { "total": 5, "committed": 3, "rollbacked
 ": 1, "running": 1, "error": 0 }, "branchTx": { "total": 5, "comm
 itted": 3, "rollbacked": 1, "running": 1, "error": 0 } }] }