

目 录

|                 |   |
|-----------------|---|
| 目录              | 1 |
| 01 背景概述         | 2 |
| 02 对外KOP鉴权      | 2 |
| 03 鉴权Token生成及使用 | 2 |

## 01 背景概述

云端录制等功能使用KOP子服务进行签名、鉴权。

## 02 对外KOP鉴权

请您先保证账号下至少存在一对密钥。由于KMR调用访问的需要，您至少需要创建一个Access Key，创建步骤如下：（1）登录金山云控制台。（2）在控制台界面，鼠标指向右上角的用户身份，然后点击Access Keys。

金山云

控制台首页

English 备案 消息 1 文档 工单 企业 费用

账号及安全

实名认证

身份与控制

Access Keys

退出登录

重要通知

您尚未进行实名认证，购买使用中国大陆服务需要先进行实名认证，立即认证

您好！

，欢迎您使用金山云控制台！

控制台使用体验调研

运维管理

产品与服务

财务管理

+

快速入口

账户总览

我的订单

我的账单

+ 添加快速入口

已开通的产品

暂无已开通的产品

项目管理

新建项目 更多

| 项目ID | 项目名称 | 成员数量 | 状态  | 操作        |
|------|------|------|-----|-----------|
| 0    | 默认项目 | 0    | 已启用 | 成员管理 账单明细 |

待办事项

0 待处理工单

0 7天内待续费

0 待支付订单

访问控制

更多

IAM安全检查：您有 5 条 安全检查未完成 查看详情

子用户登录链接 复制

0 子用户

0 用户组

0 自定义策略

0 角色

新建用户

新建自定义策略

风险预警提示

账号AccessKey具有本账号内所有产品API的最高权限，一旦泄露将导致极大的安全风险！

强烈建议您根据最小权限原则创建并使用子用户AccessKey来进行API访问和控制台操作，参看AccessKey最佳实践

继续使用

创建子用户进行管理

在AK密钥管理页面，点击新建密钥。页面关闭后密钥将无法再被查看。

注：如出现以上提示，请单击继续使用。（4）下载凭证，保存密钥，注意

生成密钥

这是唯一一次保存AK的机会，请下载凭证！

新建AccessKey成功！

查看AccessKey 详情

AccessKeyID:

密钥:

下载凭证

## 03 鉴权Token生成及使用

rtc业务 KOP域名: <http://krtc.api.ksyun.com>

生产环境方式 GO: 请在请求发送前，调用该函数进行签名。 JAVA: 对请求参数签名。

GO Token生成示例如下：

```
import (
    "github.com/aws/aws-sdk-go/aws/credentials"
    v4 "github.com/aws/aws-sdk-go/aws/signer/v4"
    "net/http"
    "strings"
    "time"
)

func addCert(r *http.Request) *http.Request {
    creds := credentials.NewStaticCredentials("conf.KopAk", "conf.KopSk", "")
    signer := v4.NewSigner(creds)

    _, err := signer.Sign(r, strings.NewReader("body"), "krtc", "cn-shanghai-1", time.Now())
    if err != nil {
        log.Debug("sign err = ", err)
    }

    return r
}
```

Java签名示例如下：

```
import com.ksc.AwsSignerV4Util;
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.HashMap;
import java.util.Map;

public class TestPost {
    public static void main(String[] args) throws Exception{
        String api = "http://10.111.94.163/cloud";
        StringBuffer response = new StringBuffer();
        String service = "krtc";
        String region = "cn-shanghai-1";
        String ak = "";
        String sk = "";
        String body = "{\n" +
            "    \"appId\": \"leel23411\",\n" +
            "    \"appName\": \"leel234\",\n" +
            "    \"uid\": \"leel234\",\n" +
            "    \"clientRequest\": { \"scene\": 1 }\n" +
            "}";

        Map<String, String> headers = new HashMap<String, String>();
        headers.put("Accept", "application/json");
        headers.put("x-action", "CloudRecordingAcquire");
        headers.put("x-version", "v1");

        try {
            URI uri = new URI(api);
            Map<String, String> h = AwsSignerV4Util.getAuthHeaderForPost(uri, body, headers, service, region, ak, sk);

            System.out.println(body);
            URL url = new URL(api);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setDoOutput(true);

            // By default it is GET request
            conn.setRequestMethod("POST");

            for (Map.Entry<String, String> entry : h.entrySet()) {
                System.out.println("Key = " + entry.getKey() + ", Value = " + entry.getValue());
                conn.setRequestProperty(entry.getKey(), entry.getValue());
            }

            DataOutputStream out = new DataOutputStream(conn.getOutputStream());
            out.writeBytes(body);
            out.flush();
            out.close();

            int responseCode = conn.getResponseCode();
            System.out.println("Sending get request : " + url);
            System.out.println("Response code : " + responseCode);

            // Reading response from input Stream
            BufferedReader in;
            if (200 <= conn.getResponseCode() && conn.getResponseCode() <= 299) {
                in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
            } else {
                in = new BufferedReader(new InputStreamReader(conn.getErrorStream()));
            }
            String output;

            while ((output = in.readLine()) != null) {
                response.append(output);
            }
            in.close();
        } catch (Exception e) {
            System.out.println(e.toString());
        }

        //printing result from response
        System.out.println(response.toString());
    }
}
```

研发测试环境 postman调试： （1）选择Authorination界面

HomeWorkspaces ▾ReportsExplore

My Workspace

NewImport

Collections

APIs

Environments

Mock Servers

Monitors

History

+

Untitled API

GETUntitled Request

×

+

...

Untitled Request

GET

▼

Enter request URL

Params

Authorization

Headers (6)

Body

P

Query Params

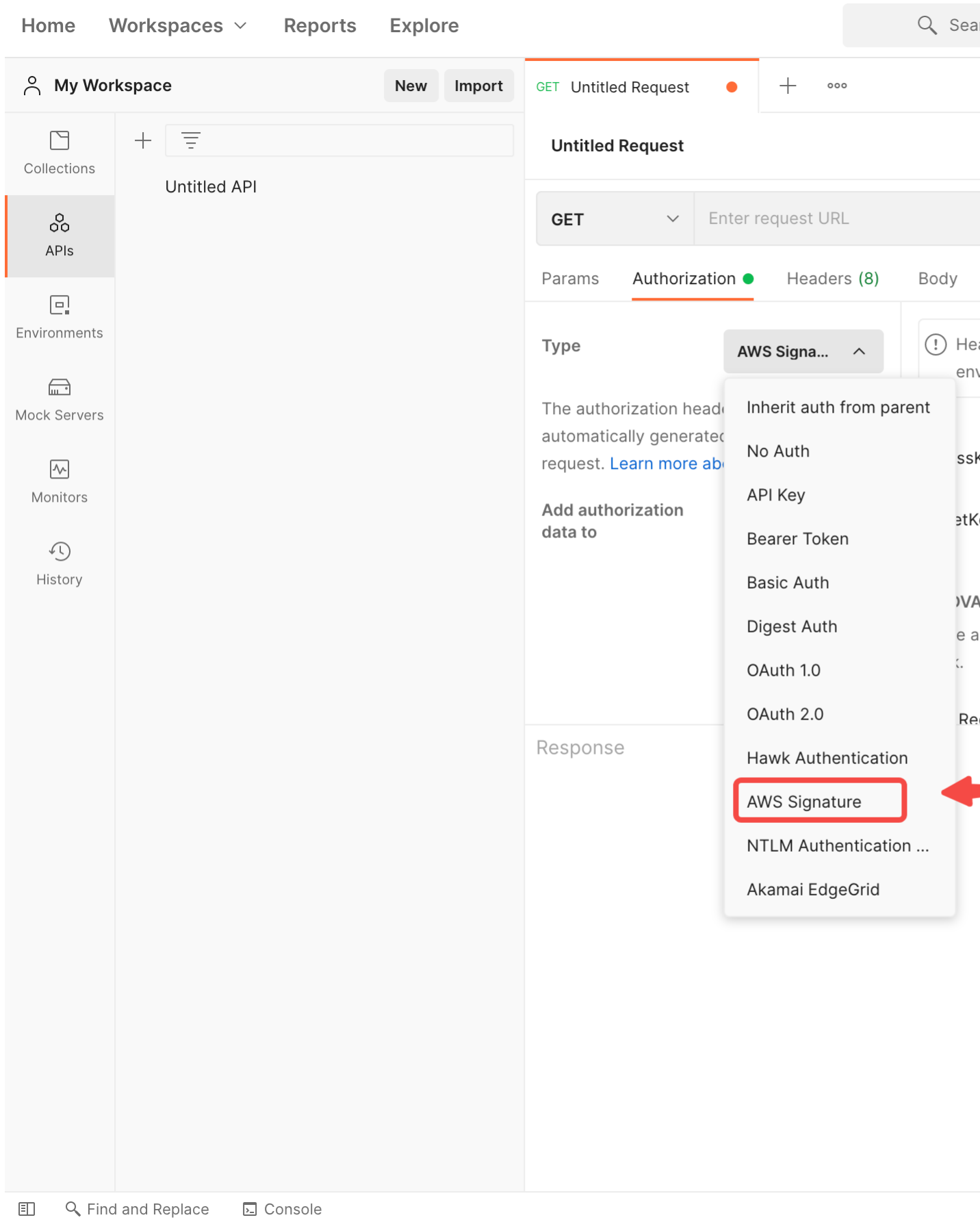
|  |     |
|--|-----|
|  | KEY |
|  | Key |

Response

Find and Replace

Console

(2) 选择AWS签名方法，填入对应的AKSK



HomeWorkspaces ▾ReportsExplore

My Workspace

NewImport

Collections

APIs

Environments

Mock Servers

Monitors

History

+

Untitled API

GETUntitled Request

+

...

Untitled Request

GET ▾Enter request URL

ParamsAuthorization ●Headers (8)Body

Type

AWS Signa... ▾

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#) ➔

Add authorization data to

Request He... ▾

! Header env

AccessK

SecretK

ADVA

These ar

blank.

AWS Rec

Response

Find and Replace

Console