

## 目录

目录	1
API概览	6
工程管理	6
日志池管理	6
日志数据	6
快速查询管理	6
告警管理	6
仪表盘管理	6
监控	6
内嵌Token	7
请求结构	7
服务地址	7
通信协议	7
请求方式	7
请求参数	7
公共响应头	7
通用错误码	7
公共参数及签名机制	8
签名机制	8
1. 创建一个正规化请求	8
2. 创建签名字符串	9
3. 计算签名信息	9
支持地域	9
CreateProject	9
请求语法	9
请求参数	10
示例	10
请求示例	10
响应示例	10
DescribeProject	10
请求语法	10
请求参数	10
响应参数	11
示例	11
请求示例	11
响应示例	11
UpdateProject	11
请求语法	11
请求参数	11
示例	12
请求示例	12
响应示例	12
DeleteProject	12
请求语法	12
请求参数	12
示例	12
请求示例	12
响应示例	13
ListProjects	13
请求语法	13

请求参数	13
响应参数	13
示例	13
请求示例	13
响应示例	14
CreateLogPool	14
请求语法	14
请求参数	14
示例	14
请求示例	14
响应示例	15
DescribeLogPool	15
请求语法	15
请求参数	15
响应参数	15
示例	15
请求示例	15
响应示例	16
UpdateLogPool	16
请求语法	16
请求参数	16
示例	16
请求示例	16
响应示例	17
DeleteLogPool	17
请求语法	17
请求参数	17
示例	17
请求示例	17
响应示例	17
ListLogPools	18
请求语法	18
请求参数	18
响应参数	18
示例	18
请求示例	18
响应示例	19
PutLogs	19
请求语法	19
请求参数	19
错误码	20
示例	20
请求示例	20
响应示例	20
GetLogs	20
请求语法	20
请求参数	20
响应参数	21
错误码	21
示例	21
请求示例	21
响应示例	21

SetIndexTemplate	22
请求语法	22
请求参数	22
示例	22
请求示例	22
响应示例	23
CreateQuickSearch	23
请求语法	23
请求参数	23
错误码	23
示例	23
请求示例	24
响应示例	24
ListQuickSearchs	24
请求语法	24
请求参数	24
响应参数	24
示例	25
请求示例	25
响应示例	25
GetQuickSearch	25
请求语法	25
请求参数	26
响应参数	26
错误码	26
示例	26
请求示例	26
响应示例	26
DeleteQuickSearchs	27
请求语法	27
请求参数	27
错误码	27
示例	27
请求示例	27
响应示例	27
CreateDashboard	28
请求语法	28
请求参数	28
响应参数	28
错误码	28
示例	28
请求示例	28
响应示例	28
DeleteDashboard	29
请求语法	29
请求参数	29
错误码	29
示例	29
请求示例	29
响应示例	29
ListDashboards	29
请求语法	29

请求参数	30
响应参数	30
示例	30
请求示例	30
响应示例	30
CreateChart	31
请求语法	31
请求参数	31
响应参数	31
示例	31
请求示例	31
响应示例	32
DeleteChart	32
请求语法	32
请求参数	32
错误码	32
示例	32
请求示例	32
响应示例	33
GetDashboardNamesByIds	33
请求语法	33
请求参数	33
响应参数	33
示例	33
请求示例	33
响应示例	33
GetChartNamesByIds	34
请求语法	34
请求参数	34
响应参数	34
示例	34
请求示例	34
响应示例	34
UpdateDashboard	35
请求语法	35
请求参数	35
示例	35
请求示例	35
响应示例	35
创建下载任务	35
请求语法	35
请求参数	36
示例	36
请求示例	36
响应示例	36
获取下载任务列表	36
请求语法	36
请求参数	37
响应参数	37
示例	37
请求示例	37
响应示例	37

获取下载链接	38
请求语法	38
请求参数	38
响应参数	38
示例	38
请求示例	38
响应示例	39
GetUsage	39
请求语法	39
请求参数	39
响应参数	39
示例	39
请求示例	40
响应示例	40
内嵌Token	40
请求语法	40
请求参数	40
示例	40
请求示例	40
响应示例	41
数据编码格式	41

# API概览

## 工程管理

接口名称	功能描述
<a href="#">CreateProject</a>	创建工程
<a href="#">DescribeProject</a>	获取工程信息
<a href="#">UpdateProject</a>	修改工程，仅支持修改描述信息
<a href="#">DeleteProject</a>	删除已有工程
<a href="#">ListProjects</a>	获取工程列表

## 日志池管理

接口名称	功能描述
<a href="#">CreateLogPool</a>	创建日志池，指定分区数等
<a href="#">DescribeLogPool</a>	获取日志池信息
<a href="#">UpdateLogPool</a>	修改日志池信息，日志池名称不支持修改
<a href="#">DeleteLogPool</a>	删除日志池
<a href="#">ListLogPools</a>	获取日志池列表

## 日志数据

接口名称	功能描述
<a href="#">PutLogs</a>	上传日志到日志池
<a href="#">GetLogs</a>	日志数据查询

## 快速查询管理

接口名称	功能描述
<a href="#">CreateQuickSearch</a>	查询语句、日志池等信息保存为快速查询
<a href="#">ListQuerySearchs</a>	获取快速查询列表
<a href="#">GetQuickSearch</a>	获取快速查询的详情
<a href="#">DeleteQuickSearchs</a>	删除快速查询，支持批量删除

## 告警管理

接口名称	功能描述
<a href="#">ValidateAlertCondition</a>	校验告警条件是否有效
<a href="#">CreateAlert</a>	创建告警，包括从搜索页面及仪表盘页面
<a href="#">DeleteAlert</a>	删除告警
<a href="#">EnableAlert</a>	开启或关闭
<a href="#">UpdateAlert</a>	修改告警

## 仪表盘管理

接口名称	功能描述
<a href="#">CreateDashboard</a>	创建仪表盘
<a href="#">DeleteDashboard</a>	删除仪表盘
<a href="#">ListDashboards</a>	仪表盘列表
<a href="#">DescribeDashboard</a>	描述仪表盘信息，列举图表名称
<a href="#">CreateChart</a>	向仪表盘添加图表
<a href="#">DeleteChart</a>	从仪表盘中删除图表
<a href="#">GetDashboardNamesByIds</a>	根据仪表盘ID获取仪表盘名称
<a href="#">GetChartNamesByIds</a>	根据图表ID获取图表名称
<a href="#">UpdateDashboard</a>	修改仪表盘信息

## 监控

接口名称	功能描述
GetUsage	获取用量数据

## 内嵌Token

接口名称	功能描述
<a href="#">GetToken</a>	获取iframe token

# 请求结构

## 服务地址

金山云日志服务KLog API的服务接入地址为：<http://klog.api.ksyun.com>

## 通信协议

支持通过HTTP或HTTPS协议进行请求通信。为保障您的服务安全性，请使用HTTPS协议进行请求通信。

## 请求方式

除GetToken接口仅支持GET外，其他接口仅支持POST(application/json)方法发送请求。

## 请求参数

金山云openAPI请求包含两类参数：公共请求参数和接口请求参数。其中，公共请求参数是每个接口都要用到的请求参数，具体可参见[公共参数及签名机制](#)小节；公共请求头是各个接口所特有的，具体见各个接口的“请求参数”描述。

# 公共响应头

日志服务API是基于HTTP协议的Rest风格接口。支持一组可以在所有API请求中使用的公共响应头，其详细说明如下：

Header名称	类型	说明
Content-Length	String	响应的 Body 长度
Content-Type	String	响应的 Body 格式，目前支持 application/json
x-klog-requestid	String	服务端针对本次请求标示的唯一 ID

# 通用错误码

当API请求发生错误的时候，服务端会返回错误信息，包括HTTP的Status Code和响应Body中的具体错误细节。其中响应Body中的错误细节为如下格式：

```
{
  "errorCode" : <ErrorCode>,
  "errorMessage" : <ErrorMessage>
}
```

通用错误码具体介绍如下表。

Http状态码	错误码 (errorCode)	错误描述 (errorMessage)	描述
411	MissingContentLength	Content-Length does not exist in http header when it is necessary	没有提供必须的 Content-Length 请求头
415	InvalidContentType	Content-Type {type} is unsupported	不支持 Content-Type 指定的类型
400	MissingContentType	Content-Type does not exist in http header when body is not empty	没有为 Body 不为空的 HTTP 请求指定 Content-Type 头
400	MissingHost	Host does not exist in http header	没有提供 HTTP 标准请求头 Host
400	MissingDate	Date does not exist in http header	没有提供 HTTP 标准请求头 Date
400	InvalidDateFormat	Date {date} must follow RFC822	Date 请求头的值不符合 RFC822 标准

400	MissingAPIVersion	x-klog-apiversion does not exist in http header	没有提供 HTTP 请求头 x-log-apiversion
400	InvalidAPIVersion	x-log-apiversion {version} is unsupported	HTTP 请求头 x-log-apiversion 的值不支持
400	MissAccessKeyId	x-klog-accesskeyid does not exist in header	没有在 Authorization 头部提供 AccessKeyId
401	Unauthorized	The AccessKeyId is unauthorized	提供的 AccessKeyId 值未授权
400	MissingSignatureMethod	x-log-signaturemethod does not exist in http header	没有提供 HTTP 请求头 x-klog-signaturemethod
400	InvalidSignatureMethod	signature method {method} is unsupported	x-klog-signaturemethod 头部指定的签名方法不支持
400	RequestTimeTooSkewed	Request time exceeds server time more than 15 minutes	请求的发送时间超过当前服务处理时间前后 15 分钟的范围
400	ProjectNotExist	Project {name} does not exist	日志工程 (Project) 不存在
400	LogPoolNotExist	LogPool {name} does not exist	日志池不存在
400	UserNotExist	User {UserId} dose not exist	用户不存在
401	SignatureNotMatch	Signature {signature} is not matched	请求的数字签名不匹配
500	InternalServerError	Internal server error message	服务器内部错误
503	ServerBusy	The server is busy, please try again later	服务器正忙, 请稍后再试

说明: 错误消息中包括{...}部分为出错相关的具体信息。例如, Project {name} does not exist中包括{name}, 表示错误消息中该部分会被具体的工程名称来替换

## 公共参数及签名机制

金山云OpenAPI支持以下两种签名算法, 您可以根据业务需要选择所使用的签名算法, 请注意两种签名算法所使用的公共参数有所区别。(1) AWS签名算法版本4, 具体可以参考[AWS文档](#) (推荐)。

- [公共参数](#)
- [签名算法](#)

(2) 简化版签名算法, 相比AWS签名算法, 签名机制更加简单。

- [公共参数](#)
- [签名算法](#)

## 签名机制

日志服务支持GET和POST两种HTTP方法, GET方法所有请求参数包括signature放置在url中, POST方法则将Signature以名为authorization header的形式放置在header中, 其主要区别在于GET方式处理的请求url长度不能过长。

签名计算的主要流程如下:

### 1. 创建一个正规化请求

在签名前, 首先将请求进行正规化格式化, 目的是让签名计算过程无二意, 其主要过程伪代码如下:

```
CanonicalRequest = HTTPRequestMethod + '\n' + CanonicalURI + '\n' + CanonicalQueryString + '\n' + CanonicalHeaders + '\n' + SignatureHeaders + '\n' + HexEncode(Hash(RequestPayload))
```

Hash指代计算哈希的算法, 目前使用SHA-256, HexEncode是对哈希值进行用16进制编码(使用小写字母)。具体步骤如下

1. 抽取HTTP请求方法(如GET、PUT、POST) 结尾附加“换行符”
2. URI绝对路径进行URI编码得到正规化URI, 如果绝对路径为空, 那么使用前斜线"/", 结尾附加“换行符”
3. 构建正规化Querystring, 结尾附加“换行符”
  - o URI编码每一个querystring参数名称和参数值(注: GET方式需要包含哈希算法、信任状、签名日期和签名header等全部参数)
  - o 按照ASCII字节顺序对参数名称严格排序
  - o 将排序号的参数名称和参数值用=连接, 按照排序结果将“参数对”用&连接
4. 构建正规化headers, 结尾附加“换行符”, 伪代码如下:

```
CanonicalHeaders = CanonicalHeadersEntry0 + CanonicalHeadersEntry1 + ... + CanonicalHeadersEntryN
```



其中：

```
CanonicalHeadersEntry = Lowercase(HeaderName) + ':' + Trimall(HeaderValue) + '\n'
```

**lowercase**表示将header名字转为小写字母，**trimall**表示去掉header值前和值后的白空格，并将header值里面的连续白空格变成单空格，但是不去掉双引号中间的任何空格，且最后的正规化headers是按照header名称排序后的结果

- 添加签名headers，结尾附加“换行符”。签名header是包含在正规化headers中名称列表，其目的是指明哪些header参与签名计算，从而忽略请求被proxy添加的额外header，其中host、x-amz-date两个header如果存在则必须添加进来，伪代码如下

```
SignedHeaders = Lowercase(HeaderName0) + ';' + Lowercase(HeaderName1) + ';' + ... + Lowercase(HeaderNameN)
```

然后处理请求body，如下

- 对请求body使用哈希算法（SHA256）计算哈希值，并将二进制哈希值结果用16进制编码表示出来（且不使用大写字符），伪代码

```
HashedPayload = Lowercase(HexEncode(Hash(requestPayload)))
```

此时，

- 将上述i-vi步骤的结果连接成一个字符串，即为正规化请求（Canonical Request）
- 将vii步的正规化请求使用vi步的哈希算法计算哈希值

## 2. 创建签名字符串

签名字符串主要包含请求以及正规化请求的元数据信息，由签名算法、请求日期、信任状和正规化请求哈希值连接组成，伪代码如下：

```
StringToSign = Algorithm + '\n' + RequestDate + '\n' + CredentialScope + '\n' + HashedCanonicalRequest
```

其中，签名算法(Algorithm)为AWS4-HMAC-SHA256，请求日期(RequestDate)格式YYYYMMDD'T' HHMMSS'Z'，信任状(CredentialScope)格式为YYYYMMDD/region/service/aws4\_request（包括请求日期（ISO 8601 基本格式）），正规化请求哈希值为上述1中第viii步的结果（注意结尾不附加“换行符”）

## 3. 计算签名信息

在计算签名前，首先从私有访问密钥（secret AccessKey）派生出签名密钥（signing key），而不是直接使用私有访问密钥；之后使用签名密钥和2中计算的签名字符串来计算签名值，具体计算过程如下

- 生成签名密钥，伪代码如下

```
kSecret = *Your KSC Secret Access Key*
kDate = HMAC("AWS4" + kSecret, Date)
kRegion = HMAC(kDate, Region)
kService = HMAC(kRegion, Service)
kSigning = HMAC(kService, "aws4_request")
```

其方式是通过HMAC算法依次生成下一个HMAC的key值（第一个为私有访问密钥字符串），而data值则依次为信任状中的各项内容（日期、region、服务、结尾字符串）；HMAC算法采用HMAC-SHA256，返回值为哈希值二进制形式（256bit，32字节），不需要做8/16进制编码显示。

- 计算签名，伪代码如下：

```
signature = HexEncode(HMAC(derived-signing-key, string-to-sign))
```

使用HMAC-SHA256算法，以签名密钥作为key，签名字符串作为data计算签名，签名后的二进制哈希值结果以16进制编码输出。

## 支持地域

地域	地域 (Region) 代码
华北1 (北京)	cn-beijing-6
俄罗斯 (莫斯科)	eu-east-1

## CreateProject

### 请求语法

```
POST /CreateProject HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.6.0
Host: klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "Description": <Description>
  "Region": <Region>
}
```

### 请求参数

CreateProject接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
Description	String	否	工程描述
Region	String	是	所属地域

### 示例

#### 请求示例

```
POST /CreateProject HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.1.0
Host: klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-project-test",
  "Description": "Description of my-project-test"
}
```

#### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

## DescribeProject

### 请求语法

```
POST /DescribeProject HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host: klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>
}
```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称

#### 响应参数

参数名称	类型	描述
CreateTime	String	创建时间
Description	String	Project 描述
UpdateTime	String	最后一次更新时间
ProjectName	String	Project 名称
Region	String	所属地域

#### 示例

##### 请求示例

```
POST /DescribeProject HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host: klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-project-test"
}
```

##### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "CreateTime": "Mon, 3 Jan 2010 08:33:47 GMT",
  "Description": "Description of my-project-test",
  "UpdateTime": "Mon, 3 Jan 2010 08:33:47 GMT",
  "ProjectName": "my-project-test",
  "Region": "cn-beijing-6"
}
```

## UpdateProject

#### 请求语法

```
POST /UpdateProject HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "Description": "Description of my-project-test"
}
```

#### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
Description	String	是	工程描述

## 示例

### 请求示例

```
POST /UpdateProject HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host: klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "projectName": "my-project-test",
  "description": "Description of my-project-test"
}
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# DeleteProject

## 请求语法

```
POST /DeleteProject HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称

## 示例

### 请求示例

```
POST /DeleteProject HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
```

```
{
  "ProjectName": "my-project-test"
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# ListProjects

## 请求语法

```
POST /ListProjects HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "Page" : <Page>,
  "Size" : <Size>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
Page	int	否	返回记录的页码，从0开始
Size	int	否	每页返回最大条目，默认 500（最大值）

## 响应参数

参数名称	类型	描述
Total	int	总project数量
Count	int	本次查询返回project数量
Projects	List	工程列表

Projects参数说明如下：

参数名称	类型	描述
CreateTime	String	创建时间
Description	String	工程描述
UpdateTime	String	最后一次修改时间
ProjectName	String	工程名称
Status	String	工程状态，Normal，Deleted
Region	String	所属地域
LogPoolNum	Int	日志池数量

## 示例

### 请求示例

```
POST /ListProjects HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
```

```
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "Page" : 0,
  "Size" : 10
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "Total" : 10,
  "Count" : 10,
  "Projects" [
    {
      "CreateTime" : "Sun, 27 May 2018 07:43:27 GMT",
      "Description" : "My test project",
      "UpdateTime" : "Sun, 27 May 2018 07:43:27 GMT",
      "ProjectName" : "my-test-project",
      "Status" : "Normal",
      "Region" : "cn-beijing-6",
      "LogPoolNum" : 2
    }
  ]
}
```

# CreateLogPool

## 请求语法

```
POST /CreateLogPool HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName" : <LogPoolName>,
  "RetentionDays" : <RetentionDays>,
  "Partitions": <Partitions>,
  "Description": <Description>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
RetentionDays	Int	是	数据保存时间，单位为天，范围1~3600
Partitions	Int	是	分区个数，2-64
Description	String	否	日志池描述

## 示例

### 请求示例

```
POST /CreateLogPool HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
```

```
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "RetentionDays": 7,
  "Partitions": 2
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# DescribeLogPool

## 请求语法

```
POST /DescribeLogPool HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <logPoolName>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称

## 响应参数

参数名称	类型	描述
ProjectName	String	工程名称
LogPoolName	String	日志池名称
CreateTime	String	创建时间
UpdateTime	String	最后一次修改时间
Description	String	日志池描述
RetentionDays	Int	数据保存时间，单位为天，范围1~3600

## 示例

### 请求示例

```
POST /DescribeLogPool HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
```

```
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool"
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "RetentionDays": 7,
  "PartitionCount": 10,
  "CreateTime": "Mon, 3 Jan 2010 08:33:47 GMT",
  "UpdateTime": "Mon, 3 Jan 2010 08:33:47 GMT",
  "Description": "This is my logPool"
}
```

# UpdateLogPool

## 请求语法

```
POST /UpdateLogPool HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
  "RetentionDays": <RetentionDays>,
  "Partitions": <Partitions>,
  "Description": <description>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
RetentionDays	Int	否	数据保存时间，单位为天，范围1~3600
Partitions	Int	否	分区数量
Description	String	否	日志池描述

## 示例

### 请求示例

```
POST /UpdateLogPool HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
```



```
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "RetentionDays": 7,
  "Partitions": 2
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# DeleteLogPool

## 请求语法

```
POST /DeleteLogPool HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称

## 示例

### 请求示例

```
POST /DeleteLogPool HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool"
}
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# ListLogPools

## 请求语法

```
POST /ListLogPools HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "Page": <Page>,
  "Size": <size>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
Page	Int	是	页码，从0开始
Size	Int	是	每页返回最大条目，默认 500（最大值）

## 响应参数

参数名称	类型	描述
ProjectName	String	工程名称
Count	Int	本次查询返回project数量
Total	Int	logPool总数
LogPools	List[LogPool]	每页返回最大条目，默认 500（最大值）

LogPools参数说明如下：

参数名称	类型	描述
CreateTime	String	创建时间
UpdateTime	String	最后一次修改时间
ProjectName	String	工程名称
LogPoolName	String	日志池名称
Partitions	Int	分区数量
RetentionDays	Int	数据保存时间，单位为天，范围1~3600
Description	String	日志池描述

## 示例

### 请求示例

```
POST /ListLogPools HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "Page": 0,
  "Size": 10
}
```

## 响应示例

```

HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName": "my-test-project",
  "Count": 5,
  "Total": 50,
  "LogPools": [
    {
      "CreateTime": "Mon, 3 Jan 2020 13:46:09 GMT",
      "UpdateTime": "Mon, 3 Jan 2020 13:46:09 GMT",
      "ProjectName": "klog project",
      "LogPoolName": "klog log pool",
      "Partitions": 5,
      "RetentionDays": 5,
      "Description": "message describe bla bla bla"
    }
  ]
}

```

# PutLogs

## 请求语法

```

POST /PutLogs?ProjectName={ProjectName}&LogPoolName={LogPoolName} HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host: <Project Endpoint>
x-klog-signature-method: hmac-sha1
x-klog-compress-type: lz4
Date: <GMT Date>
Content-Type: application/x-protobuf
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
<PB 格式日志压缩数据 LogGroup>

```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称

PB语法数据格式如下，详情请参见[数据编码格式](#)。

- Log

参数名称	类型	是否必须	描述
Time	int64	是	日志时间
Contents	List	是	每个元素为一个content，详见Content字段说明

- Contents

参数名称	类型	是否必须	描述
Key	String	是	自定义key名称
Value	String	是	自定义key对应的值

- LogGroup

参数名称	类型	是否必须	描述
Logs	List	是	Log的列表，详见Log
Filename	String	否	日志的文件名

Source String 否 日志来源，可以填写机器的ip

## 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	PostBodyInvalid	Protobuffer content cannot be parsed
400	InvalidTimestamp	Invalid timestamps are in logs
400	InvalidEncoding	Non-UTF8 characters are in logs
400	InvalidKey	Invalid keys are in logs
400	PostBodyTooLarge	Logs must be less than or equal to 3 MB and 4096 entries
400	PostBodyUncompressError	Failed to decompress logs

## 示例

### 请求示例

```
POST /PutLogs?projectName=my-project&logPoolName=my-logpool HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host: klog-cn-beijing-internal.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/x-protobuf
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
x-klog-compress-type: lz4
<PB 格式日志压缩数据 LogGroup>
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# GetLogs

## 请求语法

```
POST /GetLogs HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host: <Project Endpoint>
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称，多个日志池按逗号（,）分隔
From	Int	是	查询开始时间，unix时间戳

To	Int	是	查询结束时间，unix时间戳
Query	String	否	查询分析语法。关于查询分析的详细语法，详见查询语法，不填的情况下 返回原始日志

## 响应参数

- 原始日志：如果为查询短语（非SQL），具体返回条数以 line参数为准，如果为SQL查询，原始日志返回10条。
- 聚合日志：如果为SQL查询，需要将聚合结果返回。

参数名称	类型	描述
Time	Int	查询结果总量，如简单短语查询有150w命中，该值为1500000
Count	Int	本次查询返回结果数，如20
Logs	List	是否为SQL
Keys	List	返回所有的字段key

- Log原始日志

参数名称	类型	描述
Key1	String	key1为字段名，值为key1的值
...	...	...
Keyn	String	keyn为字段名，值为keyn的值

## 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	InvalidTimeRange	Request time range is invalid
400	InvalidQueryString	Query string is invalid
400	InvalidOffset	Offset is invalid
400	InvalidLine	line is invalid

## 示例

### 请求示例

```
POST /GetLogs HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host: klog-cn-beijing-internal.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "From": 1409529660,
  "To": 1409629660,
  "Query": "select * where key1=\"value\" limit 10"
}
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "Total": 10000,
  "Count": 1,
  "Logs": [
    {
      "__timestamp__": 1409529660,
```

```

    "__source__": "192.168.1.1",
    "Key1": "value1",
    "Key2": "value2"
  }
],
"Keys": ["key3", "key4"]
}

```

## SetIndexTemplate

### 请求语法

```

POST /SetIndexTemplate HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
  "FullTextIndex":<List(string)>,
  "IndexFields":<List(IndexField)>
}

```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
FullTextIndex	List	是	全文检索的配置项，包括区分大小写、分词符
IndexFields	List	是	字段索引配置项

- FullTextIndex参数

参数名称	类型	是否必须	描述
Lowercase	boolean	是	区分大小写
Separator	String	是	分词符

- IndexFields参数

参数名称	类型	是否必须	描述
FieldName	String	是	字段名称
FieldType	String	是	数据类型, 包括TEXT、LONG、DOUBLE、DATE类型
FieldAlias	String	否	字段别名
Lowercase	Boolean	否	区别大小写, 当数据类型是text时, 需配置该项
Separator	List	否	分词符, 当数据类型是text时, 需配置该项

### 示例

#### 请求示例

```

POST /SetIndexTemplate HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host: klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": "testproject",

```

```

"LogPoolName": "testlogpool",
"FullTextIndex": {
  "Lowercase": true,
  "Separator": " ";=() [] {} ?@&<>/:\n\t\r"
},
" IndexFields ": [
  {
    "FieldName": "log",
    "FieldType": "TEXT",
    "FieldAlias": "appLog",
    "Lowercase": true,
    "Separator": " ";=() [] {} ?@&<>/:\n\t\r"
  }
]
}

```

## 响应示例

```

HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322

```

# CreateQuickSearch

## 请求语法

```

POST /CreateQuickSearch HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <logPoolName>,
  "QuickSearchName": <QuickSearchName>,
  "Description": <description>,
  "TimeRange": <timeRange>,
  "Query": <query>
}

```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
QuickSearchName	String	是	快速查询名称
Description	String	否	快速查询的备注
TimeRange	String	是	查询数据的时间范围
Query	String	是	查询语句

## 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	QuickSearchAlreadyExist	quickSearch {quickSearchName} already exists
400	QuickSearchInfoInvalid	quicksearch info is invalid

## 示例

## 请求示例

```
POST /CreateLogPool HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "QuickSearchName": "my-test-search",
  "Description": "This is my quickSearch",
  "TimeRange": "last 15 minutes",
  "Query": "select * where key1=\"value\" limit 10"
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# ListQuickSearchs

## 请求语法

```
POST /ListQuickSearchs HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
  "Filter": <Filter>,
  "Page": <page>,
  "Size": <size>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
Filter	String	否	根据快速查询名称过滤列表，支持模糊搜索
Page	Int	否	页码，从0开始
Size	Int	否	每页返回最大条目，默认 500（最大值）

## 响应参数

参数名称	类型	描述
ProjectName	String	工程名称
LogPoolName	String	日志池名称
Count	Int	本次查询返回project数量
Total	Int	logPool总数



QuickSearchs array[QuickSearch] 每页返回最大条目，默认 500（最大值）

QuickSearch参数说明如下：

参数名称	类型	描述
QuickSearchName	String	快速查询名称
Description	String	快速查询备注
CreateTime	String	创建时间
LogPoolName	String	日志池名称
TimeRange	String	查询数据的时间范围
RetentionDays	Int	数据保存时间，单位为天，范围1~3600
Query	String	查询语句

## 示例

### 请求示例

```
POST /ListuickSearchs HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-LogPool",
  "Filter": "log pool",
  "Page": 0,
  "Size": 10
}
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "Filter": "search",
  "Count": 5,
  "Total": 50,
  "QuickSearchs": [
    {
      "QuickSearchName": "my-test-search",
      "Description": "message describe bla bla bla",
      "CreateTime": "Mon, 3 Jan 2020 13:46:09 GMT",
      "LogPoolName": "my-test-logpool",
      "TimeRange": "last 15 minutes",
      "Query": "select * where key1=\"value\" limit 10"
    }
  ]
}
```

## GetQuickSearch

### 请求语法

```
POST /GetQuickSearch HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
```

```

Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
  "QuickSearchName": <QuickSearchName>
}

```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
QuickSearchName	String	是	快速查询名称

## 响应参数

参数名称	类型	描述
ProjectName	String	工程名称
QuickSearchName	String	快速查询名称
Description	String	备注
CreateTime	String	创建时间
LogPoolName	String	日志池名称
TimeRange	String	查询数据的时间范围
Query	String	查询语句

## 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	ProjectNotExsit	project {projectName} does not exist
400	LogPoolNotExsit	logPool {logPoolName} does not exist

## 示例

### 请求示例

```

POST /GetQuickSearch HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "QuickSearchName": "my-test-search"
}

```

### 响应示例

```

HTTP/1.1 200Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName": my-test-project,
  "LogPoolName": my-test-logpool,
  "QuickSearchName": "my-test-search",
  "Description": "This is my quickSearch",
}

```

```

"CreateTime": "Mon, 3 Jan 2010 08:33:47 GMT",
"TimeRange": "last 15 minutes",
"Query": "select * where key1=\"value\" limit 10"
}

```

## DeleteQuickSearchs

### 请求语法

```

POST /DeleteQuickSearchs HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "LogPoolName": <LogPoolName>,
  "QuickSearchNames" : <QuickSearchNames>
}

```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
LogPoolName	String	是	日志池名称
QuickSearchName list		是	快速查询名称

### 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	ProjectNotExsit	project {projectName} does not exist
400	LogPoolNotExsit	logPool {logPoolName} does not exist

### 示例

#### 请求示例

```

POST /DeleteQuickSearch HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "LogPoolName": "my-test-logpool",
  "QuickSearchNames" :["myquicksearch001"]
}

```

#### 响应示例

```

HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322

```

# CreateDashboard

## 请求语法

```
POST /CreateDashboard HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "DashboardName" <DashboardName>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
DashboardName	String	是	仪表盘名称

## 响应参数

参数名称	类型	描述
DashboardId	String	仪表盘ID

## 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	DashboardAlreadytExist	Dashboard {dashboardName} already exist
400	LogPoolNotExsit	logPool{logPooltName} does not exist

## 示例

### 请求示例

```
POST /CreateDashboard HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "DashboardName" : "my-dashboard"
}
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "DashboardId" : "A7967D81EFF5E3CD447FB6D8DF294E20"
```

```
}
```

## DeleteDashboard

### 请求语法

```
POST /DeleteDashboard HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "DashboardId": <DashboardId>
}
```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
DashboardId	String	是	仪表盘ID

### 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	DashboardNotExist	Dashboard {dashboardId} not exist

### 示例

#### 请求示例

```
POST /DeleteDashboard HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "DashboardId": "A7967D81EFF5E3CD447FB6D8DF294E20"
}
```

#### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

## ListDashboards

### 请求语法

```
POST /ListDashboards HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
```

```
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "Offset" : <offset>,
  "Size" : <size>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
Page	Int	否	返回的分页数，从0开始
Size	Int	否	每页返回最大条目，默认 500（最大值）

## 响应参数

参数名称	类型	描述
Total	Int	仪表盘总数
Dashboards	list	每页返回最大条目，默认 500（最大值）
Count	Int	返回的仪表盘数量
ProjectName	String	工程名称

Dashboard参数说明如下：

参数名称	类型	描述
DashboardName	String	仪表盘名称
DashboardId	String	仪表盘ID
CreateTime	Date	创建时间

## 示例

### 请求示例

```
POST /ListDashboards HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project"
}
```

### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName" : "my-test-project",
  "Count" : 5,
  "Total" : 50,
  "Dashboards": [
    {
      "DashboardName": "my-dashboard",
      "DashboardId": "dashboard-12535235-001",
```

```

    "CreateTime": "2020-02-02 15:30: 00"
  }
]
}

```

## CreateChart

### 请求语法

```

POST /CreateChart HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "DashboardId": <DashboardId>,
  "ChartName": "<ChartName>",
  "ChartType": "table",
  "Search": {
    "LogPoolName": "my-logPool",
    "TimeRange": "last 15 minutes",
    "Query": "select count(*) as pv where code=500"
  },
  "Display": {
    .....
  }
}

```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
DashboardId	String	是	仪表盘ID
ChartName	String	否	图表名称
ChartType	String	否	图表类型，line，table
Search	Search	是	查询主体
Display	Display	是	展示参数

- Search

参数名称	类型	是否必须	描述
LogPoolName	String	是	日志池名称
TimeRange	String	是	查询数据的时间范围
Query	String	是	查询语句

### 响应参数

参数名称	类型	描述
ChartId	String	图表ID

### 示例

#### 请求示例

```

POST /CreateChart HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80

```

```
Connection: Keep-Alive
{
  "DashboardId": <dashboardId>,
  "ChartName": "my-chart",
  "ChartType": "table",
  "Search": {
    "LogPoolName": "my-logPool",
    "TimeRange": "last 15 minutes",
    "Query": "select count(*) as pv where code=500"
  },
  "Display": {
    ..... //待定
  }
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ChartId": "chart-123124243-001"
}
```

# DeleteChart

## 请求语法

```
POST /DeleteChart HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ChartId": <ChartId>
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ChartId	String	是	图表ID

## 错误码

除公共错误码外，还可能返回如下特有错误码：

Http状态码	errorCode	errorMessage
400	ChartNotExist	Chart {chartId} not exist

## 示例

### 请求示例

```
POST /DeleteChart HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
```



```
Connection: Keep-Alive
{
  "ChartId": "chart-1502342324-001"
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

# GetDashboardNamesByIds

## 请求语法

```
POST /GetDashboardNameById HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "DashboardIds": [<id1>,<id2>...]
}
```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
DashboardIds	List	是	仪表盘id列表

## 响应参数

参数名称	类型	描述
ProjectName	String	工程名称
DashboardNames	List	仪表盘名称

## 示例

### 请求示例

```
POST /GetDashboardNamesByIds HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "DashboardIds": ["dashboard-158965429-1"]
}
```

### 响应示例

```
HTTP/1.1 200
```

```
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName": "my-test-project",
  "DashboardNames": [
    "my-dashboard"
  ]
}
```

## GetChartNamesByIds

### 请求语法

```
POST /GetChartNamesByIds HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "ProjectName": <ProjectName>,
  "CharIds": [<id1>, <id2>...]
}
```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
ProjectName	String	是	工程名称
ChartIds	List	是	图表id列表

### 响应参数

参数名称	类型	描述
ProjectName	String	工程名称
ChartNames	List	图表名称

### 示例

#### 请求示例

```
POST /GetChartNamesByIds HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "ProjectName": "my-test-project",
  "ChartIds": ["chart-1502342324-001", "chart-1502342324-002"]
}
```

#### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
```

```
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "ProjectName": "my-test-project",
  "ChartNames": ["my-chart-1", "my-chart-2"]
}
{
  "ProjectName": "my-test-project",
  "DashboardNames": [
    "my-dashboard"
  ]
}
```

## UpdateDashboard

### 请求语法

```
POST /UpdateDashboard HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "DashboardId": <DashboardId>,
  "DashboardName": <DashboardName>
}
```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
DashboardId	String	是	仪表盘ID
DashboardName	String	是	仪表盘名称

### 示例

#### 请求示例

```
POST /UpdateDashboard HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "DashboardId": "dashboard-158965429-1",
  "DashboardName": "my-dashboard-test"
}
```

#### 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
```

## 创建下载任务

### 请求语法

```

POST /CreateDownloadTask HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "From": 0,
  "To": 1632712166000, //毫秒值
  "LogPoolName": "logPoolName",
  "ProjectName": "projectName"
}

```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
From	int64	是	日志起始时间
To	int64	是	日志截止时间，时间跨度不能超过24小时
LogPoolName	String	是	日志池名称
ProjectName	String	是	工程名称

### 示例

#### 请求示例

```

POST /CreateDownloadTask HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "From": 0,
  "To": 1632712166000, // 毫秒值
  "LogPoolName": "logPoolName",
  "ProjectName": "projectName"
}

```

#### 响应示例

```

HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322

```

## 获取下载任务列表

### 请求语法

```

POST /ListDownloadTasks HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "Page": 0,

```

```

    "Size": 50,
    "ProjectName": "projectName",
  }

```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
Page	int64	是	页码
Size	int64	是	每页大小
ProjectName	String	是	工程名称

## 响应参数

参数名称	类型	描述
Toal	int	任务总数
Count	int	当前返回任务数
Downloads	array[Download]	每页返回最大条目，默认 500（最大值）

Downloads参数说明如下：

参数名称	类型	描述
DownloadID	String	任务ID
From	int64	查询起始时间戳
To	int64	查询结束时间戳
LogPoolName	String	日志池名称
Status	String	任务状态

## 示例

### 请求示例

```

POST /ListDownloadTasks HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "Page": 0,
  "Size": 50,
  "ProjectName": "projectName",
}

```

### 响应示例

```

HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "Total": 100,
  "Count": 10,
  "Downloads": [
    {
      "DownloadID": "56129339-0650-4206-8e21-badc36cd657c",
      "From": 0,
      "To": 1632712166000,
      "LogPoolName": "logpool1",
      "Status": "Done"
    },
    {
      "DownloadID": "102a6b59-0650-4206-8e21-badc36cd657c",

```

```
    "From": 0,  
    "To": 1632712166000,  
    "LogPoolName": "logpool1",  
    "Status": "Done"  
  }  
]  
}
```

## 获取下载链接

### 请求语法

```
POST /GetDownloadUrls HTTP/1.1  
Authorization: <AuthorizationString>  
x-klog-api-version: 0.1.0  
Host:klog.api.ksyun.com  
x-klog-signature-method: hmac-sha1  
Date: <GMT Date>  
Content-Type: application/json  
Content-MD5: <Content-MD5>  
Content-Length: <ContentLength>  
Connection: Keep-Alive  
{  
  "DownloadID": "56129339-0650-4206-8e21-badc36cd657c",  
  "ProjectName": "projectname"  
}
```

### 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。参数详情具体如下表。

参数名称	类型	是否必须	描述
DownloadID	String	是	任务下载ID，通过获取下载任务列表获取
ProjectName	String	是	工程名称

### 响应参数

参数名称	类型	描述
Urls	array[Url]	返回的Url，每个文件最大500M，超过按多条Url返回

Url参数说明如下：

参数名称	类型	描述
From	int64	查询起始时间戳
To	int64	查询结束时间戳
Url	String	日志Url，可下载
FileName	String	文件名称
Status	String	日志状态
Size	Float	日志大小
SizeUnit	String	日志大小单位

## 示例

### 请求示例

```
POST /ListDownloadTasks HTTP/1.1  
Authorization: <AuthorizationString>  
x-klog-api-version: 0.1.0  
Host:klog.api.ksyun.com  
x-klog-signature-method: hmac-sha1  
Date: <GMT Date>  
Content-Type: application/json  
Content-MD5: <Content-MD5>  
Content-Length: <ContentLength>  
Connection: Keep-Alive  
{  
  "DownloadID": "56129339-0650-4206-8e21-badc36cd657c",  
  "ProjectName": "projectname"  
}
```

## 响应示例

```

HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "Urls": [
    {
      "From": 1632702166000,
      "To": 1632712166000,
      "Url": "https://asdf/102a6b59-1632712166000.gz",
      "FileName": "102a6b59-1632712166000.gz",
      "Status": "done",
      "Size": 1.1,
      "SizeUnit": "GB"
    }
  ]
}

```

# GetUsage

## 请求语法

```

POST /GetUsage HTTP/1.1
Authorization: <AuthorizationString>
x-klog-api-version: 0.1.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  Projects: List<string> //工程列表
}

```

## 请求参数

该接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
Projects	List	否	工程名称列表
Metrics	List	否	指标名称列表，不传返回所有；WriteFlow, ReadFlow, Storage, SearchRequests, WriteRequests
From	String	否	开始时间戳
To	String	否	结束时间戳

## 响应参数

参数详情具体如下表。

参数名称	类型	描述
Data	List	图表名称
TimeSeries	List	时间戳序列

- Metric

参数名称	类型	描述
MetricName	String	指标名称
Output	List	指标列表，每个时间戳一个
YesterdayUsage	int64	昨日数据
DayToDayRatio	int64	日环比

## 示例

## 请求示例

```
POST /GetChartNamesByIds HTTP/1.1
Authorization: KLOG <yourAccessKeyId>:<yourSignature>
x-klog-api-version: 0.6.0
Host:klog.api.ksyun.com
x-klog-signature-method: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "Projects": ["my-project1", "my-project2"],
  "Metrics":["writeFlow", "readFlow"],
  "From": "2020-06-15 17:25:40",
  "To": "2020-06-25 17:25:40"
}
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "TimeSeries" : [1589277626, 1589277626, 1589277626]
  "Data" : [
    {
      "MetricName": "writeFlow",
      "YesterdayUsage": 13089,
      "DayToDayRatio": 1.5,
      "Output" [1000, 10000, 23231, 123123]
    },
    {
      "MetricName": "inFlow",
      "YesterdayUsage": 13089,
      "DayToDayRatio": 1.5,
      "Output" [1000, 10000, 23231, 123123]
    }
  ]
}
```

# 内嵌Token

## 请求语法

```
https://klog.api.ksyun.com/?
Action=GetToken&Version=2020-07-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKLTGo0pHK-EQWiDZWTSBS112Q%2F20160914%2Fen-beijing-6%2Fiam%2Faws4_request
&X-Amz-Date=20160914T114902Z
&X-Amz-SignedHeaders=host
&X-Amz-Signature=88f6284257863dedfc350da05d19d07f76cca622e93b829f5ce26c1a75d3da39
&ExpireTime=20
```

## 请求参数

ExpireTime接口无特有请求头。获取更多信息，请参见公共请求头。 参数详情具体如下表。

参数名称	类型	是否必须	描述
ExpireTime	String	否	ExpireTime单位为分钟，不填则默认token过期时间为30分钟

## 示例

### 请求示例

```
GET /GetToken HTTP/1.1
https://klog.api.ksyun.com/?
Action=GetToken&Version=2020-07-31
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKLTGo0pHK-EQWiDZWTSBS112Q%2F20160914%2Fen-beijing-6%2Fiam%2Faws4_request
```



```
&X-Amz-Date=20160914T114902Z
&X-Amz-SignedHeaders=host
&X-Amz-Signature=88f6284257863dedfc350da05d19d07f76cca622e93b829f5ce26c1a75d3da39
&Expires=20
```

## 响应示例

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-klog-requestid: 5B0A619F205DC3F30EDA9322
{
  "Token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOiJlMzE4NTMyOTEsInN1YiI6ImFjY291bnRlZDogMjMzLn0. jTLYz_jQJ4LOB-cX_OVXkZ-8DmL2gEgyr7srHj58_LRQ"
  "Expires": 1631850328
}
```

## 数据编码格式

日志服务支持使用Protocol Buffer格式作为标准的日志写入格式。

Protocol Buffer格式用于结构化数据交换格式，当用户需要写入日志时，需要把原始日志数据序列化成如下格式的Protocol Buffer数据流，然后才能通过API写入服务端。

```
message Log
{
  message Content
  {
    required string key = 1; // 每组字段的 key
    required string value = 2; // 每组字段的 value
  }
  required int64 time = 1; // 时间戳，UNIX时间格式
  repeated Content contents = 2; // 一条日志里的多个kv组合
}
message LogGroup
{
  repeated Log logs = 1; // 多条日志合成的日志数组
  optional string reserved = 2; // 目前暂无效用
  optional string filename = 3; // 日志文件名
  optional string source = 4; // 日志来源，一般使用机器IP
}
message LogGroupList //主要用于结果返回用，PullLogs
{
  repeated LogGroup logGroupList = 1; // 日志组列表
}
```

## 说明

- 在使用Protobuf时要保证Key-Value对的唯一性，否则会出现行为未定义的错误。
- 关于Protocol Buffer格式的更多信息请参见[Github首页](#)。
- 关于日志服务写入日志的API的详细描述，请参见[Putlogs](#)。