Log Service

API Reference

MORE THAN JUST CLOUD | Alibaba Cloud

API Reference

Overview

Log Service (Log) is a platform service specific to logs. Log supports real-time collection, storage and delivery of various types of logs. Besides, Log synchronizes data between tables for the MaxCompute and ships logs to the MaxCompute for analysis.

Besides operations on logs through the management console, Log allows you to use Application Programming Interfaces (APIs) to write and query logs, and manage your projects and LogStores. Currently, the following APIs are available.

Object	Method	
Log	A basic concept of Log	
Config	List, Create, Delete, Get, and Update	
	GetAppliedMachineGroups (query the machine group applied)	
MachineGroup	List, Create, Delete, Get, and Update	
	Apply/Remove (apply/remove an application)	
	GetAppliedConfigs (query the list of configurations applied)	
LogStore	List, Create, Delete, Get, and Update	
	GetLogs (query logs), and GetHistograms (query log distribution)	
Shard	List, Split, Merge, and Delete	
	PostLogStoreLogs (write a log)	
	GetCursor (locate the log location)	
	PullLogs (consume a log)	

Shipper	GetShipperStatus (query the status of a LogShipper task)	
	RetryShipperTask (retry a failed LogShipper task)	

You can use the APIs to perform the following operations:

- Collect logs based on configuration and machine group.
- Create a LogStore, write a LogStore, and read a log.
- Set access control rules for different users.

Note:

- Currently, APIs support the Rest style.
- To use the APIs, you need to know the API access address.
- Security verification is required for all requests from APIs. Refer to Request signature to see API request signature mechanism and procedures.
- Similar to general cloud accounts, subaccounts of RAM, STS or RAM can use their AK signatures to call APIs on Log. The STS temporary identity consists of a temporary AK and a special HTTP header which needs to participate in the signature. For details, refer to Documentation.

Service endpoint

Public network endpoint

The Log Service endpoint is a URL for accessing a project and its internal log data. It is associated with the project’ s Alibaba Cloud region and project name. Currently, Log Service has been activated in multiple Alibaba Cloud regions. The endpoint for the public network in each region is listed as follows:

Region	Endpoint
China East 1 (Hangzhou)	cn-hangzhou.log.aliyuncs.com
China East 2 (Shanghai)	cn-shanghai.log.aliyuncs.com
China East 1 (Hangzhou-AntCloud, but the access through the public network is not supported)	cn-hangzhou-finance-intranet.log.aliyuncs.com
China North 1 (Qingdao)	cn-qingdao.log.aliyuncs.com
China North 2 (Beijing)	cn-beijing.log.aliyuncs.com
China North 3 (Zhangjiakou)	cn-zhangjiakou.log.aliyuncs.com

China South 1 (Shenzhen)	cn-shenzhen.log.aliyuncs.com
Hong Kong	cn-hongkong.log.aliyuncs.com
Asia Pacific NE 1 (Tokyo)	ap-northeast-1.log.aliyuncs.com
Asia Pacific SE 1 (Singapore)	ap-southeast-1.log.aliyuncs.com
Asia Pacific SE 2 (Sydney)	ap-southeast-2.log.aliyuncs.com
Middle East 1 (Dubai)	me-east-1.log.aliyuncs.com
US West 1 (Silicon Valley)	us-west-1.log.aliyuncs.com
EU Central 1 (Frankfurt)	eu-central-1.log.aliyuncs.com

When accessing a specific project, you need to give a final access address composed of a project name and the region where the project is located. The specific format is as follows:

```
<project_name>.<region_endpoint>
```

For example, if the project name is big-game and it is in the China East 1 (Hangzhou) region, the endpoint would be:

```
big-game.cn-hangzhou.log.aliyuncs.com
```

When creating a Log Service project, you must specify a certain region. Once specified, the region cannot be changed and you cannot migrate the project across regions. After creating a project, you must select a root endpoint address that matches the region to compose the access address for this project. The endpoint will be used for API requests.

Classic network endpoint

If you are using Log Service API in an Alibaba Cloud ECS instance, you can also use the Intranet endpoint (using this to access Log Service will not consume ECS public network traffic and will save valuable ECS public bandwidth). The Log Service Intranet root endpoint for each region is as follows:

Region	Root Endpoint
China East 1 (Hangzhou)	cn-hangzhou-intranet.log.aliyuncs.com
China East 2 (Shanghai)	cn-shanghai-intranet.log.aliyuncs.com
China East 1 (Hangzhou-AntCloud)	cn-hangzhou-finance-intranet.log.aliyuncs.com
China North 1 (Qingdao)	cn-qingdao-intranet.log.aliyuncs.com
China North 2 (Beijing)	cn-beijing-intranet.log.aliyuncs.com
China South 1 (Shenzhen)	cn-shenzhen-intranet.log.aliyuncs.com

China South 1 (Shenzhen-AntCloud)	No classic network virtual machine exists. The VPC network endpoint is used.
Hong Kong	cn-hongkong-intranet.log.aliyuncs.com
US West 1 (Silicon Valley)	us-west-1-intranet.log.aliyuncs.com

For the above example, the Intranet endpoint is as follows:

```
big-game.cn-hangzhou-intranet.log.aliyuncs.com
```

VPC network endpoint

If you use VPC network for the ECS instances, you can use the VPC region endpoint for Log Service. The Log Service VPC root endpoint for each region is as follows:

Region	Root Endpoint
China East 1 (Hangzhou)	cn-hangzhou-vpc.log.aliyuncs.com
China East 2 (Shanghai)	cn-shanghai-vpc.log.aliyuncs.com
China North 1 (Qingdao)	cn-qingdao-vpc.log.aliyuncs.com
China North 2 (Beijing)	cn-beijing-vpc.log.aliyuncs.com
China North 3 (Zhangjiakou)	cn-zhangjiakou-vpc.log.aliyuncs.com
China South 1 (Shenzhen)	cn-shenzhen-vpc.log.aliyuncs.com
China South 1 (Shenzhen-AntCloud)	cn-shenzhen-finance-vpc.log.aliyuncs.com
Hong Kong	cn-hongkong-vpc.log.aliyuncs.com
Asia Pacific NE 1 (Tokyo)	ap-northeast-1-vpc.log.aliyuncs.com
Asia Pacific SE 1 (Singapore)	ap-southeast-1-vpc.log.aliyuncs.com
Asia Pacific SE 2 (Sydney)	ap-southeast-2-vpc.log.aliyuncs.com
Middle East 1 (Dubai)	me-east-1-vpc.log.aliyuncs.com
US West 1 (Silicon Valley)	us-west-1-vpc.log.aliyuncs.com
EU Central 1 (Frankfurt)	eu-central-1-vpc.log.aliyuncs.com

For the above example, the VPC network endpoint is as follows:

```
big-game.cn-hangzhou-vpc.log.aliyuncs.com
```

Currently, the preceding Log Service API endpoints only support HTTP protocol.

Access Key

Alibaba Cloud Access Key is a “secure password” designed to secure the access to your cloud resources over APIs (instead of the management console). You can use the Access Key to sign API request content in order to pass server security authentication.

This Access Key is generated and used by pairing an AccessKeyId and AccessKeySecret. Each Alibaba Cloud user can create multiple access key/secret key pairs. You can activate (Active), deactivate (Inactive), or delete generated access key/secret key pairs at will.

You can create and manage all access key/secret key pairs through the Key Management Page on the Alibaba Cloud console. You need to save the access key/secret key pairs because they are a critical factor in Alibaba Cloud’s API request security authentication. If a key pair may have been leaked, it is recommended that you immediately delete this key pair and generate a new one.

Public request header

Log Service APIs are HTTP-based interfaces in the REST style. The Log Service API supports a set of public request headers that can be used in all API requests (unless stated otherwise, each Log Service API request must provide these public request headers). The detailed definitions are given below.

Header name	Type	Description
Accept	String	Type of public request headers that the client expects from the server. Currently, application/json and application/x-protobuf are supported. This field is optional and valid only for GET requests. For the specific value, see the specific interface’s definition.
Accept-Encoding	String	Compression algorithm that the client expects from the server. Currently, options include LZ4, DEFLATE or empty (no compression). This field is optional and valid only for GET requests. For the specific value, see the specific interface’s definition.
Authorization	String	Signature content. For more details, refer to Request signature.

Content-Length	Numeric Value	Length of the HTTP request Body defined in RFC 2616. If the request has no Body, it is not necessary to provide this request header.
Content-MD5	String	String produced after the request Body undergoes MD5 computation, results in uppercase. If the request has no Body, it is not necessary to provide this request header.
Content-Type	String	Type of the HTTP request Body defined in RFC 2616. Currently, Log Service API requests support only application/x-protobuf. If the request has no Body, it is not necessary to provide this request header. For the specific value, see the specific interface’s definition.
Date	String	The current time when sent. Currently, the parameter supports only the RFC 822 format, and the GMT standard time is used. The formatted string is as follows: %a, %d %b %Y %H:%M:%S GMT (for example: Mon, 3 Jan 2010 08:33:47 GMT).
Host	String	Complete HOST name of the HTTP request (does not include http:// and other such protocol headers), for example, big-game.cn-hangzhou.sls.aliyuncs.com.
x-log-apiversion	String	API version. The current version is 0.6.0.
x-log-bodyrawsize	Numeric value	Initial size of the request Body. When the request has no Body, the value is 0. When the Body is a compressed data, the value is the size of the initial data before compression. The value range for this field is 0 to 3x1024x1024. This field is optional and needs to be filled only when the Body is compressed.

x-log-compresstype	String	Compression type of the Body in the API request. Currently, LZ4 and DEFLATE are supported (RFC 1951. For details about the zlib format, refer to RFC 1950). If the Body is not compressed, it is not necessary to provide this request header.
x-log-date	String	The current time when sent. The format is consistent with that of the Date header. This request header is optional. If a request contains this public request header, this value will replace the standard Date header value in sever request authentication. No matter whether there is an x-log-date header, the standard HTTP Date header must be provided.
x-log-signaturemethod	String	Signature calculation method. Currently, only hmac-sha1 is supported.
x-acs-security-token	String	Use the STS temporary identity to send data. It is required only when the STS temporary identity is used to send data.

Note:

- The maximum difference between the time expressed in a request’ s Date header and the time the server receives the request is 15 minutes. If this difference exceeds 15 minutes, the server will reject this request. If the request contains an x-log-date header, the time difference is calculated based on the value of the x-log-date header.
- If a compression algorithm is specified (in x-log-compresstype) for the request, the initial data needs to be compressed and then put in the HTTP Body. The Content-Length and Content-MD5 header length are calculated based on the size of the compressed Body.
- When HTTP requests are sent from certain platforms, they cannot specify a Date header (the sending time is specified automatically in the platform’ s internal database). Therefore, the correct Date value cannot be used to calculate the request signature. In this case, specify an x-log-date header and use the value of this header for request signature calculation. After receiving an API request, the Log Service server will first determine whether the request contains an x-log-date header. If any, the server uses the header value for signature authentication; otherwise, the server uses the HTTP standard header Sate for signature authentication.

Public response header

Log Service APIs are HTTP-based interfaces in the REST style. All Log Service API responses provide a group of public response headers. The detailed definitions are as follows.

Header name	Type	Description	
Content-Length	Numeric value	Length of the HTTP response content defined in RFC 2616.	
Content-MD5	String	MD5 value of the HTTP response content defined in RFC 2616. It is an uppercased string produced after the request body undergoes MD5 computation.	
Content-Type	String	Type of the HTTP response content defined in RFC 2616. Currently, responses of application/json and application/x-protobuf types are supported by the Log Service server.	
Date	String	The current time when sent. Currently, the parameter supports only the RFC 822 format, and the GMT standard time is used. The formatted string is as follows: %a, %d %b %Y %H:%M:%S GMT (for example: Mon, 3 Jan 2010 08:33:47 GMT).	
x-log-requestid	String	Unique ID generated by the server that marks this request. This response header is not related to actual applications. It is	

		mainly used for tracking and investigating problems. For API request failure troubleshooting purpose, you can provide this ID to the Log Service team.	

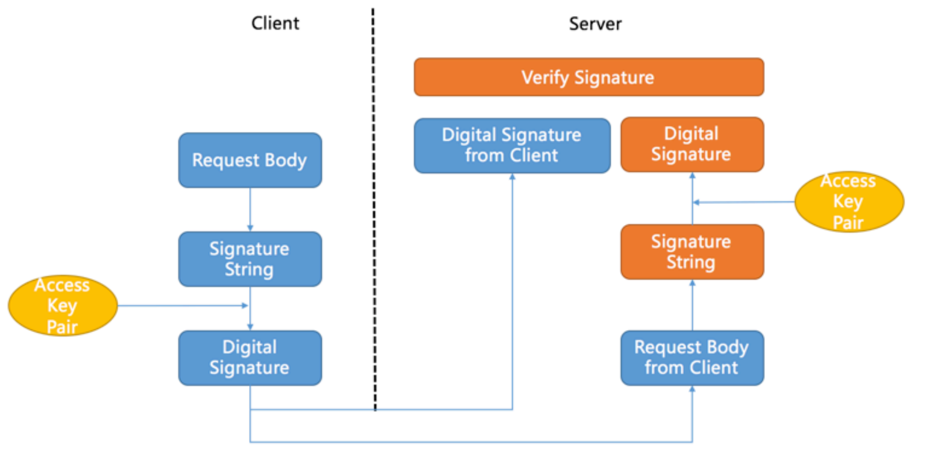
Request authentication

To ensure security of users’ log data, all HTTP requests of the Log Service API must undergo security authentication. Currently, this security authentication is completed through a symmetric encryption algorithm based on Alibaba Cloud’ s Access key.

This process is as follows:

1. The requester generates a SignString based on the API request content (including the HTTP Header and Body).
2. The requester uses Alibaba Cloud’ s access key pair (AccessKeyID and AccessKeySecret) to sign the signature string generated in the first step, forming a digital signature for this API request
3. The requester sends both the API request content and digital signature to the server.
4. After receiving the request, the server repeats steps 1 and 2 (**Note:** the server will retrieve the user access key pair used by this request in the background) and calculates the expected digital signature for this request on the server
5. The server compares the expected digital signature to the digital signature sent with the request. If they are completely consistent, the request passes security authentication. Otherwise, the request is immediately rejected.

This entire process can be intuitively described by the following diagram:



The security authentication process above can also be used for the following purposes:

- Confirm which user sends the API request. Because a user is required to designate the key pair to generate the digital signature before sending the request, the server can use this key pair to confirm the user’ s identity and then perform access permission management.
- Confirm if a user’ s request has been tampered with during network transmission. Because the server will recalculate the digital signature of the received request content, if this content has been tampered with during transmission, the digital signature will not match.

Sign an API request

In order to pass API request security authentication, you must sign this API request in the client (in other words, generate a correct digital signature) and use the HTTP header “Authorization” to transmit the request’ s digital signal over the network. The specific format of the Authorization header is as follows:

```
Authorization:LOG <AccessKeyId>:<Signature>
```

As shown in the above format, the Authorization header value contains the AccessKeyID from the user’ s key pair and the corresponding AccessKeySecret will be used in the construction of the signature value. The following provides a detailed explanation of how to construct this signature value.

Step One: Prepare a suitable Alibaba Cloud access key pair.

As described above, to generate a digital signature for an API request, you must use an access key pair (AccessKeyId/AccessKeySecret). You may use an existing access key pair or create a new one. The key must be in the “Active” state.

Step Two: Generate the request’ s signature string.

The signature string of Log Service API is generated with Method, Header and Body of the HTTP request. The following describes how to generate a signature string.

```
SignString = VERB + "\n"
+ CONTENT-MD5 + "\n"
+ CONTENT-TYPE + "\n"
+ DATE + "\n"
+ CanonicalizedLOGHeaders + "\n"
+ CanonicalizedResource
```

In the above formula, \n indicates the newline escape character and + (plus sign) indicates the string concatenation operator. The other parts are defined as follows:

Name	Definition	Example
VERB	Name of the HTTP request method	PUT, GET, POST and so on
CONTENT-MD5	MD5 value of the HTTP request Body, which must be an uppercased string	875264590688CA6171F6228 AF5BBB3D2
CONTENT-TYPE	Type of the HTTP request Body	application/x-protobuf
DATE	Standard time stamp header of the HTTP request (follows RFC 1123 format and uses the GMT standard time)	Mon, 3 Jan 2010 08:33:47 GMT
CanonicalizedLOGHeaders	String constructed by custom headers prefixed by x-log and x-acs in the HTTP request (for the specific construction method, see the description below)	x-log-apiversion:0.6.0\nx-log-bodyrawsize:50\nx-log-signaturemethod:hmac-sha1
CanonicalizedResource	String constructed by the HTTP request resources (for the specific construction method, see the description below)	/logstores/app_log

In some HTTP requests without the Body, the CONTENT-MD5 and CONTENT-TYPE fields are null strings respectively. Described below is how to generate a SignString in such case:

```
SignString = VERB + "\n"
+ "\n"
+ "\n"
+ DATE + "\n"
+ CanonicalizedLOGHeaders + "\n"
+ CanonicalizedResource
```

As described in Public request headers, the custom request header x-log-date is introduced in Log Service API. If you specify this header in your request, the header value will replace the value of the HTTP standard request header Date in signature calculation.

The CanonicalizedLOGHeaders construction method is as follows:

1. Convert the names of all HTTP request headers prefixed with x-log and x-acs to lowercase letters.
2. All LOG custom request headers obtained in the previous step are sorted alphabetically in ascending order.
3. Any space separators at either end of the request headers and content are deleted.
4. Separate all the headers and content using the \n separator to form the final CanonicalizedLOGHeader.

The CanonicalizedResource construction method is as follows:

1. Set CanonicalizedResource to an empty string (“”).
2. Enter the LOG resource to be accessed, for example, /logstores/logstorename. The field is left blank when there is no logstorename.
3. If the request contains a query string (QUERY_STRING), then add ？ and the query string at the end of the CanonicalizedResource string.

The QUERY_STRING is the alphabetized string of the request parameters in the URL. An = (equals sign) is used between the names and values of these parameters to form a string. Then, the parameter name-value pairs are alphabetically sorted and connected with the & symbol to form a string. This formula is illustrated below:

```
QUERY_STRING = "KEY1=VALUE1" + "&" + "KEY2=VALUE2"
```

Step Three: Generate the request’ s digital signature

Currently, Log Service API only supports one digital signature algorithm, namely the default digital signature algorithm hmac-sha1. The entire signature formula is as follows:

```
Signature = base64(hmac-sha1(UTF8-Encoding-Of(SignString)，AccessKeySecret))
```

> HMAC-SHA1 defined in RFC/rfc2104 of is used as the signature method.The AccessKeySecret used in the above formula must correspond to the AccessKeyID used in the Authorization header. Otherwise, the request will not be able to pass authentication by the server.

After the digital signature value is calculated, use the value to construct a complete security authentication header for the Log Service API request in the Authorization header format described at the top of this section, and enter the security authentication header in the HTTP request. Then, the HTTP request can be sent.

Examples of the request authentication process

The following two examples demonstrate the entire process of request authentication. First, assume your key pair for Log Service API signature is as follows:

```
AccessKeyId = "bq2sjzesjmo86kq35behupbq"
AccessKeySecret = "4fdO2fTDDnZPU/L7CHNdemB2Nsk="
```

Example 1:

You need to send a GET request as follows to list all LogStores in the ali-test-project. The HTTP request is as follows:

```
GET /logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host: ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

The following signature string is generated for the Log Service API request:

```
GET\n\n\nMon, 09 Nov 2015 06:11:16 GMT\nx-log-apiversion:0.6.0\nx-log-signaturemethod:hmac-
sha1\n/logstores?logstoreName=&offset=0&size=1000
```

Because a GET request has no HTTP Body, the value of CONTENT-TYPE and CONTENT-MD5 in the generated signature string is an empty string respectively. If the AccessKeySecret specified previously is used to calculate a signature, the resulting signature will be:

```
jEYOTCJs2e88o+y5F4/S5IsnBJQ=
```

Finally, the digitally signed HTTP request content sent is as follows:

```
GET /logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host: ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Authorization: LOG bq2sjzesjmo86kq35behupbq:jEYOTCJs2e88o+y5F4/S5IsnBJQ=
```

Example 2:

You need to write the follow log into the LogStore named test-logstore in the **ali-test-project** project in Example 1.

```
topic=""
time=1447048976
source="10.230.201.117"
"TestKey": "TestContent"
```

Therefore, according to the Log Service API definition, the following HTTP request should be constructed:

```
POST /logstores/test-logstore HTTP/1.1
Date: Mon, 09 Nov 2015 06:03:03 GMT
Host: test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Content-MD5: 1DD45FA4A70A9300CC9FE7305AF2C494
Content-Length: 52
x-log-apiversion:0.6.0
x-log-bodyrawsize:50
x-log-compresstype:lz4
x-log-signaturemethod:hmac-sha1

<Log contents serialized in a ProtoBuffer format byte stream>
```

In this HTTP request, the written log content is first serialized in a ProtoBuffer format (refer to ProtoBuffer format for more details) and then used as the request Body. Therefore, the Content-Type header value of this request is application/x-protobuf. Similarly, the Content-MD5 header value is the MD5 value of the request Body. According to the above signature string construction method, the signature string corresponding to this request is:

```
POST\n1DD45FA4A70A9300CC9FE7305AF2C494\napplication/x-protobuf\nMon, 09 Nov 2015 06:03:03 GMT\nx-log-apiversion:0.6.0\nx-log-bodyrawsize:50\nx-log-compresstype:lz4\nx-log-signaturemethod:hmac-sha1\n/logstores/test-logstore
```

In the same way, if the AccessKeySecret specified previously is used to calculate a signature, the resulting signature will be:

```
XWLGYHGg2F2hcfxWxMLiNkGki6g=
```

Finally, the digitally signed HTTP request content sent is as follows:

```
POST /logstores/test-logstore HTTP/1.1
Date: Mon, 09 Nov 2015 06:03:03 GMT
Host: test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Content-MD5: 1DD45FA4A70A9300CC9FE7305AF2C494
Content-Length: 52
x-log-apiversion:0.6.0
x-log-bodyrawsize:50
x-log-compresstype:lz4
x-log-signaturemethod:hmac-sha1
Authorization: LOG bq2sjzesjmo86kq35behupbq:XWLGYHGg2F2hcfxWxMLiNkGki6g=

<Log contents serialized in a ProtoBuffer format byte stream>
```

Common error codes

When an API request error occurs, the server returns an error message, including the HTTP status code and the specific error details in the response body. The error details in the response body are formatted as follows:

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

Of all error messages that may be returned by the server, some are applicable for the majority of APIs and others are unique to certain APIs. The following table lists the common error codes that will appear in the response of multiple APIs. The special error codes of an API are described in the reference to that particular API.

HTTP Status Code	Error Code	Error Message	Description
411	MissingContentLength	Content-Length does not exist in http header when it is necessary.	The required Content-Length request header has not been provided.
415	InvalidContentType	Content-Type {type} is unsupported.	The specified Content-Type is not supported.
400	MissingContentType	Content-Type does not exist in http header when body is not empty.	The Content-Type header is not specified when the HTTP request body is not empty.
400	MissingBodyRawSize	x-log-bodyrawsize does not exist in header when it is necessary.	The required x-log-bodyrawsize request header has not been provided in the compression scenario.
400	InvalidBodyRawSize	x-log-bodyrawsize is invalid.	The x-log-bodyrawsize value is invalid.
400	InvalidCompressType	x-log-compresstype {type} is unsupported.	The compress type specified in x-log-compresstype is not supported.
400	MissingHost	Host does not exist in http header.	The HTTP standard request header Host has not been provided.
400	MissingDate	Date does not exist	The HTTP standard

		in http header.	request header Date has not been provided.
400	InvalidDateFormat	Date {date} must follow RFC822.	The Date request header value does not comply with the RFC822 standard.
400	MissingAPIVersion	x-log-apiversion does not exist in http header.	The HTTP request header x-log-apiversion has not been provided.
400	InvalidAPIVersion	x-log-apiversion {version} is unsupported.	The value of the HTTP request header x-log-apiversion is not supported.
400	MissAccessKeyId	x-log-accesskeyid does not exist in header.	No AccessKeyId has been provided in the Authorization header.
401	Unauthorized	The AccessKeyId is unauthorized.	The provided AccessKeyId value is unauthorized.
400	MissingSignatureMethod	x-log-signaturemethod does not exist in http header.	The HTTP request header x-log-signaturemethod has not been provided.
400	InvalidSignatureMethod	signature method {method} is unsupported.	The signature method specified by the x-log-signaturemethod header is not supported.
400	RequestTimeTooSkewed	Request time exceeds server time more than 15 minutes.	The request sending time is more than 15 minutes before or after the time of processing by the server.
404	ProjectNotExist	Project {name} does not exist.	The log project does not exist.
401	SignatureNotMatch	Signature {signature} is not matched.	The request’s digital signature does not match.
403	WriteQuotaExceed	Write quota is exceeded.	Exceeds the log write quota.
403	ReadQuotaExceed	Read quota is exceeded.	Exceeds the log read quota.

500	InternalServerError	Internal server error message.	Internal server error.
503	ServerBusy	The server is busy, please try again later.	The server is busy, please try again later.

The {...} contained in the error message indicates the specific error information. For example, the ProjectNotExist error message contains {name}. This indicates that, in the error message, this section will be replaced by the specific project name.

LogStore related interfaces

CreateLogstore

Creates a logstore for the project.

Example:

```
POST /logstores
```

Request syntax

```
POST /logstores HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

{
"logstoreName" : <logStoreName>,
"ttl": <ttl>,
"shardCount": <shardCount>
}
```

Request parameters

Attribute Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name, which must be

			unique in the project.
ttl	integer	Yes	The data retention period (days).
shardCount	integer	Yes	The shard count for this logstore.

Request header

The CreateLogstore interface does not have a specific request header. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

The CreateLogstore interface does not have a specific response header. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

The system returns the HTTP status code 200.

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP Status Code	Error Code	Error Message
400	LogstoreAlreadyExist	logstore {logstoreName} already exist
500	InternalServerError	Specified Server Error Message
400	LogstoreInfoInvalid	logstore info is invalid
400	ProjectQuotaExceed	Project Quota Exceed

Detailed description

The logstore cannot be created if quota is invalid.

Example

Request example:

```
POST /logstores HTTP/1.1
Header :
{
```

```
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:8IwDTWugRK1AZAo0dWQYpffhy48=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 07:35:00 GMT,
Content-Length=55,
x-log-signaturemethod=hmac-sha1,
Content-MD5=7EF43D0B8F4A807B95E775048C911C72,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
Body :
{
"logstoreName": "test-logstore",
"ttl": 1,
"shardCount": 2
}
```

Response example:

```
HTTP/1.1 200 OK
Header:
{
Date=Wed, 11 Nov 2015 07:35:00 GMT,
Content-Length=0,
x-log-requestid=5642EFA499248C827B012B39,
Connection=close,
Server=nginx/1.6.1
}
```

DeleteLogstore

Deletes a logstore, including all shard data and indexes.

Request syntax

```
DELETE /logstores/{logstoreName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name, which is unique in the same project.

Request header

The DeleteLogstore interface does not have a specific request header. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

The DeleteLogstore interface does not have a specific response header. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

The system returns the HTTP status code 200.

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP Status Code	Error Code	Error Message
404	LogStoreNotExist	logstore {logstoreName} not exist
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
DELETE /logstores/test_logstore HTTP/1.1
Header :
{
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:fPsNBIuJR1xvQZolwi8+Cw5R/fQ=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 08:09:38 GMT,
Content-Length=0,
x-log-signaturemethod=hmac-sha1,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
```

Response example:

```
HTTP/1.1 200 OK
Body:
{
```

```
Date=Wed, 11 Nov 2015 08:09:39 GMT,
Content-Length=0,
x-log-requestid=5642F7C399248C817B013A07,
Connection=close,
Server=nginx/1.6.1
}
```

UpdateLogstore

Updates LogStore attributes. Currently, only Time To Live (TTL) and shard attributes can be updated.

Request syntax

```
PUT /logstores/{logstoreName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

{
"logstoreName": <logstoreName>,
"ttl": <ttl>,
"shardCount": <shardCount>
}
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name, which is unique in the same project.
ttl	integer	Yes	The life cycle of log data, ranging from 1 to 365 days.
shardCount	integer	Yes	The number of shards, ranging from 1 to 10.

Request header

The UpdateLogstore interface does not have a special request header. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

The UpdateLogstore interface does not have a special response header. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

The system returns the HTTP status code 200.

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	Project {ProjectName} not exist
404	LogStoreNotExist	logstore {logstoreName} not exist
400	LogStoreAlreadyExist	logstore {logstoreName} already exist
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	invalid shard count,you can only increase the count

Detailed description

Currently, shards can be added rather than deleted.

Example

Request example:

```
PUT /logstores/test-logstore HTTP/1.1
Header:
{
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:wFcl3ohVJupCi0ZFxRD0x4IA68A=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 08:28:19 GMT,
Content-Length=55,
x-log-signaturemethod=hmac-sha1,
Content-MD5=757C60FC41CC7D3F60B88E0D916D051E,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
Body :
{
"logstoreName": "test-logstore",
"ttl": 1,
"shardCount": 2
}
```

Response example:

```
HTTP/1.1 200 OK
Header:
{
Date=Wed, 11 Nov 2015 08:28:20 GMT,
Content-Length=0,
x-log-requestid=5642FC2399248C8F7B0145FD,
Connection=close,
Server=nginx/1.6.1
}
```

GetLogstore

Views LogStore attributes.

Request syntax

```
GET /logstores/{logstoreName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name, which is unique in the same project.

Request header

The GetLogstore interface does not have a special request header. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

The GetLogstore interface does not have a special response header. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

The system returns the HTTP status code 200.

```
{
  “logstoreName" : <logstoreName>,
"ttl": <ttl>,
"shardCount": <shardCount>,
"createTime": <createTime>,
"lastModifyTime": <lastModifyTime>
}
```

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP Status Code	Error Code	Error Message
404	ProjectNotExist	Project {ProjectName} not exist
404	LogstoreNotExist	logstore {logstoreName} not exist
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
GET /logstores/test-logstore HTTP/1.1
Header :
{
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:6ga/Cvj51rFatX/DtTkcQB/CALk=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 07:53:29 GMT,
Content-Length=0,
x-log-signaturemethod=hmac-sha1,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
```

Response example:

```
HTTP/1.1 200 OK
```

```
Header :
{
Date=Wed, 11 Nov 2015 07:53:30 GMT,
Content-Length=107,
x-log-requestid=5642F3FA99248C817B01352D,
Connection=close,
Content-Type=application/json,
Server=nginx/1.6.1
}
Body :
{
  “logstoreName" : test-logstore,
"ttl": 1,
"shardCount": 2,
"createTime": 1447833064,
"lastModifyTime": 1447833064

}
```

ListLogstore

On the ListLogstores interface, lists names of all LogStores under a specified project.

Request syntax

```
GET /logstores HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
offset(optional)	integer	No	The starting position of a returned record, 1 by default.
size(optional)	integer	No	The maximum number of entries returned per page, 500 by default (maximum).
logstoreName	string	No	The LogStore name used to filter (partial matching supported).

Request header

No special request header is available. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

No special response header is available. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

When the ListLogStores request is successful, the response body will contain a list of names of all LogStores under a specified project. The format is as follows:

Name	Type	Description
count	Integer	The number of returned LogStores.
total	Integer	The total number of LogStores.
logstores	String array	The list of names of returned LogStores.

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	Project {ProjectName} not exist
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	Invalid parameter size, (0.6.0]
400	InvalidLogStoreQuery	logstore Query is invalid

Example

Request example:

```
GET /logstores HTTP/1.1
Header:
{
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:we34Siz/SBVyVGMGmMDnvp0xSPo=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 08:09:39 GMT,
Content-Length=0,
x-log-signaturemethod=hmac-sha1,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
```

Response example:

```
HTTP/1.1 200 OK
Header:
{
Date=Wed, 11 Nov 2015 08:09:39 GMT,
Content-Length=52,
x-log-requestid=5642F7C399248C8D7B01342F,
Connection=close,
Content-Type=application/json,
Server=nginx/1.6.1
}
Body:
{
"count":1,
"logstores":["test-logstore"],
"total":1
}
```

ListShards

Lists all available shards in the LogStore.

Request syntax

```
GET /logstores/<logstorename>/shards HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	No	LogStore name

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

Content-Type: application/json

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

It is an array consisting of shards.

```
[
{
"shardID":0
  “status”： “readwrite",
  “inclusiveBeginKey”："00000000000000000000000000000000",
  “exclusiveEndKey”："80000000000000000000000000000000",
  “createTime”：1453949705
},
{
...
},
{
...
}
]
```

Detailed description

N/A

Specific error codes

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	LogStoreNotExist	logstore {logstoreName} not exist
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

The {name} variable in the above error messages will be replaced by the actual logstore name.

Example

Request example:

```
GET /logstores/sls-test-logstore/shards
Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Thu, 12 Nov 2015 03:40:31 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:xEOsJ3xeivcRq0GbvACiO37jH0I="
}
```

Response example:

```
Header:
{
"content-length": "57",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Thu, 12 Nov 2015 03:40:31 GMT",
"content-type": "application/json",
"x-log-requestid": "56440A2F99248C050600C74C"
}
Body :
[
{
  “shardID” : 1,
  “status” :  “readwrite",
  “inclusiveBeginKey” : "00000000000000000000000000000000",
  “exclusiveEndKey” : "8000000000000000000000000000000",
  “createTime” :1453949705
},
{
  “shardID” : 2,
  “status” :  “readwrite",
  “inclusiveBeginKey” : "80000000000000000000000000000000",
  “exclusiveEndKey” : "ffffffffffffffffffffffffffffffff",
  “createTime” :1453949705
},
{
  “shardID” : 0,
  “status” :  “readonly",
  “inclusiveBeginKey” : "00000000000000000000000000000000",
  “exclusiveEndKey” : "ffffffffffffffffffffffffffffffff",
  “createTime” :1453949705
}

]
```

SplitShard

Splits a specified shard in readwrite status.

Request syntax

```
POST /logstores/<logstorename>/shards/<shardid>?action=split&key=<splitkey> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name
shardid	int	Yes	shard ID
splitkey	string	Yes	Split location

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

Content-Type: application/json

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

In an array consisting of 3 shards, the first shard is not split, and the other two are the results of splitting.

```
[
{
"shardID":33
  “status” :  “readonly",
  “inclusiveBeginKey” : "ee000000000000000000000000000000",
  “exclusiveEndKey” : "ffffffffffffffffffffffffffffffff",
  “createTime” :1453949705
},
{
"shardID":163
  “status” :  “readwrite",
  “inclusiveBeginKey” : "ee000000000000000000000000000000",
  “exclusiveEndKey” : "ef000000000000000000000000000000",
  “createTime” :1453949705
},
{
"shardID":164
  “status” :  “readwrite",
  “inclusiveBeginKey” : "ef000000000000000000000000000000",
  “exclusiveEndKey” : "ffffffffffffffffffffffffffffffff",
  “createTime” :1453949705
}
]
```

Detailed description

N/A

Specific error codes

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	LogStoreNotExist	logstore {logstoreName} not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	invalid mid hash
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

The name variable in the above error messages will be replaced by the actual LogStore Name.

Example

Request example:

```
POST /logstores/logstorename/shards/33?action=split&key=ef000000000000000000000000000000
Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou.sls.aliyuncs.com",
"Date": "Thu, 12 Nov 2015 03:40:31 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:xEOsJ3xeidfdgRq0GbvACiO37jH0I="
}
```

Response example:

```
Header:
{
"content-length": "57",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Thu, 12 Nov 2015 03:40:31 GMT",
"content-type": "application/json",
"x-log-requestid": "56440A2F99248C050600C74C"
}
Body :
[
{
"shardID":33
  "status" :  "readonly",
  "inclusiveBeginKey" : "ee000000000000000000000000000000",
  "exclusiveEndKey" : "ffffffffffffffffffffffffffffffff",
  "createTime" :1453949705
},
{
"shardID":163
  "status" :  "readwrite",
  "inclusiveBeginKey" : "ee000000000000000000000000000000",
  "exclusiveEndKey" : "ef000000000000000000000000000000",
  "createTime" :1453949705
},
{
"shardID":164
  "status" :  "readwrite",
  "inclusiveBeginKey" : "ef000000000000000000000000000000",
  "exclusiveEndKey" : "ffffffffffffffffffffffffffffffff",
  "createTime" :1453949705
}
]
```

MergeShards

Merges two adjacent shards in readwrite status. When a shard ID is specified in parameters, the server will automatically find the next adjacent shard.

Request syntax

```
POST /logstores/<logstorename>/shards/<shardid>?action=merge HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name
shardid	int	Yes	shard ID

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

Content-Type: application/json

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

In an array consisting of 3 shards, the first shard is the result of merging, and the other two are not merged.

```
[
{
'shardID': 167,
'status': 'readwrite',
'inclusiveBeginKey': 'e0000000000000000000000000000000',
'createTime': 1453953105,
'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
},
{
'shardID': 30,
'status': 'readonly',
'inclusiveBeginKey': 'e0000000000000000000000000000000',
'createTime': 0,
'exclusiveEndKey':
'e7000000000000000000000000000000'
},
{
```

```
'shardID': 166,
'status': 'readonly',
'inclusiveBeginKey': 'e7000000000000000000000000000000',
'createTime': 1453953073,
'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
}
]
```

Detailed description

N/A

Specific error codes

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	LogStoreNotExist	logstore {logstoreName} not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	can not merge the last shard
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

The name variable in the above error messages will be replaced by the actual LogStore Name.

Example

Request example:

```
POST /logstores/logstorename/shards/30?action=merge
Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou.sls.aliyuncs.com",
"Date": "Thu, 12 Nov 2015 03:40:31 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:xEOsJ3xeidfdgRq0GbvACiO37jH0I="
}
```

Response example:

```
Header:
{
"content-length": "57",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Thu, 12 Nov 2015 03:40:31 GMT",
"content-type": "application/json",
"x-log-requestid": "56440A2F99248C050600C74C"
}
Body :

[
{
'shardID': 167,
'status': 'readwrite',
'inclusiveBeginKey': 'e0000000000000000000000000000000',
'createTime': 1453953105,
'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
},
{
'shardID': 30,
'status': 'readonly',
'inclusiveBeginKey': 'e0000000000000000000000000000000',
'createTime': 0,
'exclusiveEndKey':
'e7000000000000000000000000000000'
},
{
'shardID': 166,
'status': 'readonly',
'inclusiveBeginKey': 'e7000000000000000000000000000000',
'createTime': 1453953073,
'exclusiveEndKey': 'ffffffffffffffffffffffffffffffff'
}
]
```

DeleteShard

Deletes a readonly shard.

Request syntax

```
DELETE /logstores/<logstorename>/shards/<shardid> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name
shardid	int	Yes	Shard ID

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

Content-Type: application/json

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Detailed description

N/A

Specific error codes

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	LogStoreNotExist	logstore {logstoreName} not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	only readonly shard id can be deleted
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

The {name} variable in the above error messages will be replaced by the actual logstore name.

Example

Request example:

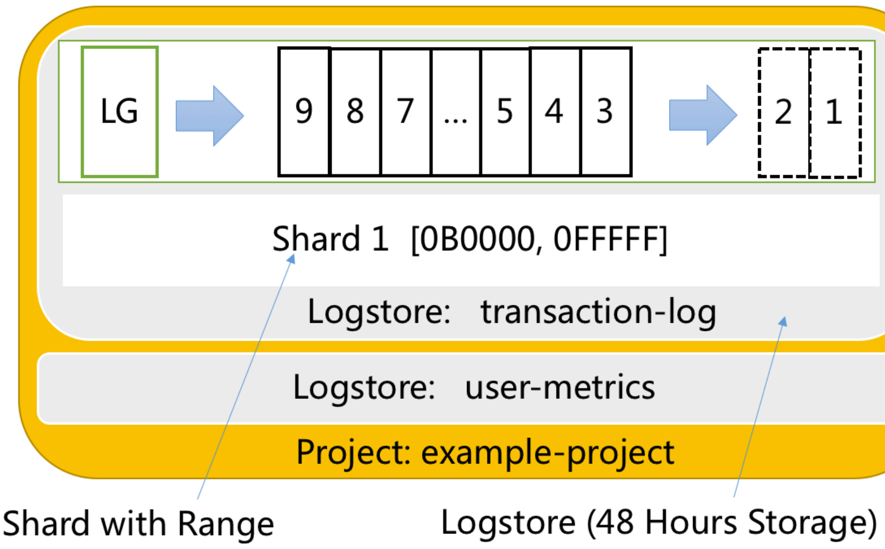
```
DELETE /logstores/logstorename/shards/30
```

```
Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou.sls.aliyuncs.com",
"Date": "Thu, 12 Nov 2015 03:40:31 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:xEOsJ3xeidfdgRq0GbvACiO37jH0I="
}
```

GetCursor

The GetCursor is used to get the cursor based on the time. The following diagram shows the relationship between the cursor and the project, LogStore, and shard:

- A project has multiple logstores.
- Each logstore can have multiple shards.
- A cursor can be used to get the location of the specified log.



Request syntax

```
GET /logstores/ay42/shards/2?type=cursor&from=1402341900 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
```

Request parameters

Parameter Name	Type	Required or Not	Description

shard	string	Yes	
type	string	Yes	Cursor
from	string	Yes	Time point (in UNIX seconds), or begin, end

LogStore lifecycle:

The lifecycle of a logstore is specified by the lifeCycle field in the attribute. For example, it is assumed that the current time is 2015-11-11 09:00:00 and the lifecycle is 24. Therefore, the consumable data time section in each shard is [2015-11-10 09:00:00, 2015-11-11 09:00:00]. The time here is the server time.

Using the parameter “from”, you can locate the logs within the lifecycle in the shard. Assuming that the logstore lifecycle is [begin_time,end_time) and the parameter “from” is set to from_time, then:

```
 from_time <= begin_time or from_time == "begin" : Returns the cursor location corresponding to begin_time.
from_time >= end_time or from_time == "end" : Returns the cursor location for writing the next entry at the current
time point (no data at this cursor location currently).
from_time > begin_time and from_time < end_time : Returns the cursor location for the first data packet whose
receipt time at the server is >= from_time.
```

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

```
{
"cursor": "MTQ0NzI5OTYwNjg5NjYzMjM1Ng=="
}
```

Detailed description

N/A

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message

404	LogStoreNotExist	Logstore {Name} not exist
400	ParameterInvalid	Parameter From is not valid
400	ShardNotExist	Shard {ShardID} not exist
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	the logstore has no shard

Example

Request example:

```
GET /logstores/sls-test-logstore/shards/0?type=cursor&from=begin
Header:
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Thu, 12 Nov 2015 03:56:57 GMT",
"x-log-apiversion": "0.6.0",
"Content-Type": "application/json",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:+vo0Td6PrN0CGoskJoOiAsnkXgA="
}
```

Response example:

```
Header:
{
"content-length": "41",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Thu, 12 Nov 2015 03:56:57 GMT",
"content-type": "application/json",
"x-log-requestid": "56440E0999248C070600C6AA"
}
Body:
{
"cursor": "MTQ0NzI5OTYwNjg5NjYzMjM1Ng=="
}
```

PullLogs

Obtains logs based on the cursor and quantity. You must specify a shard when the system obtains logs. In the Storm scenario, elective and collaborative consumption can be performed through LoghubClientLib. Currently, only [log group lists in PB format] can be read.

Request syntax

```
GET /logstores/ay42/shards/0?type=logs&cursor=MTQ0NzMyOTQwMTEwMjEzMDkwNA==&count=100 HTTP/1.1
Accept: application/x-protobuf
Accept-Encoding: lz4
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Parameter Name	Type	Required or Not	Description
type	string	Yes	Logs
cursor	string	Yes	A start point for data reading
count	int	Yes	The number of returned log groups, ranging from 0 to 1000

Request header

- Accept: application/x-protobuf
- Accept-Encoding: LZ4, deflate, or “”

For details about the public request header of the API, refer to Public Request Header.

Response header

- x-log-cursor： The next cursor of the currently read data
- x-log-count： The number of returned logs

For details about the public response header of the API, refer to Public Response Header.

Response element

Data from serialized logs in PB format (which may be compressed).

Detailed description

N/A

Specific error codes

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	LogStoreNotExist	Logstore {Name} not exist
400	ParameterInvalid	Parameter Cursor is not valid
400	ParameterInvalid	ParameterCount should be in [0-1000]
400	ShardNotExist	Shard {ShardID} not exist
400	InvalidCursor	this cursor is invalid
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
Read data on the shard numbered 0.

GET /logstores/sls-test-
logstore/shards/0?cursor=MTQ0NzMyOTQwMTEwMjEzMDkwNA==&count=1000&type=log

Header:
{
"Authorization"="LOG 94to3z418yupi6ikawqqd370:WeMYZp6bH/SmWEgryMrLhbxK+7o=",
"x-log-bodyrawsize"=0,
"User-Agent" : "sls-java-sdk-v-0.6.0",
"x-log-apiversion" : "0.6.0",
"Host" : "ali-test-project.cn-hangzhou-failover-intranet.sls.aliyuncs.com",
"x-log-signaturemethod" : "hmac-sha1",
"Accept-Encoding" : "lz4",
"Content-Length": 0,
"Date" : "Thu, 12 Nov 2015 12:03:17 GMT",
"Content-Type" : "application/x-protobuf",
"accept" : "application/x-protobuf"
}
```

Response example:

```
Header:
{
"x-log-count" : "1000",
"x-log-requestid" : "56447FB20351626D7C000874",
"Server" : "nginx/1.6.1",
"x-log-bodyrawsize" : "34121",
"Connection" : "close",
"Content-Length" : "4231",
"x-log-cursor" : "MTQ0NzMyOTQwMTEwMjEzMDkwNA==",
"Date" : "Thu, 12 Nov 2015 12:01:54 GMT",
"x-log-compresstype" : "lz4",
"Content-Type" : "application/x-protobuf"
}

Body:
Data from the compressed <log group list in PB format>
```

Page flip

To flip the page (to get the next token) without returning data, the system can send HTTP HEAD requests.

PostLogStoreLogs

Writes log data in a specified LogStore. Currently, only log groups in PB format can be written. There are two writing modes:

- Load balancing: The system automatically writes all writable shards in the LogStore in load balancing mode. This method is highly available for writing (SLA: 99.95%), applicable to scenarios in which data writing and consumption are independent of shards, for example, non-reordering.
- Key hashing: A key is needed for data writing. The server will automatically write shards within the key range. For example, the system can hash a producer (for example, an instance) to a fixed shard based on the name to ensure orderly writing and consumption on the shard (it can ensure that during merging or splitting a key appears only on one shard at a time point, refer to Shard Data Model).

Request syntax

Load balancing

```
POST /logstores/<logstorename>/shards/lb HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/x-protobuf
Content-Length: <Content Length>
Content-MD5: <Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-bodyrawsize: <BodyRawSize>
x-log-compresstype: lz4
x-log-signaturemethod: hmac-sha1
```

```
<Compressed log data in PB format>
```

Key hasing

The system adds an x-log-hashkey to the header to determine which shard the key falls on. This parameter is optional. If you leave it blank, the system will automatically switch to the load balancing mode.

```
POST /logstores/<logstorename>/shards/lb HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/x-protobuf
Content-Length: <Content Length>
Content-MD5: <Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-bodyrawsize: <BodyRawSize>
x-log-compresstype: lz4
x-log-hashkey : 14d2f850ad6ea48e46e4547edbbb27e0
x-log-signaturemethod: hmac-sha1

<Compressed log data in PB format>
```

Request parameters

Name	Type	Required	Description
logstorename	String	Yes	The LogStore name of the logs to write.

Request header

In key hashing mode, an x-log-hashkey request header is needed (refer to the preceding example). For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

After the request is successful, there are no response elements.

Detailed description

- A maximum of 3 MB or 4,096 logs can be written on the PutLogs interface. Once the logs written in exceed 3MB or 4096 lines, the entire request fails, and no logs are written into the

LogStore.
- The server checks the format of the logs written on the PutLogs interface. (For log formats, refer to Core Concept). Once any log does not comply with the specification, the entire request fails, and no logs are written.=- The server checks time stamps of the Logs written on the PutLogs interface each time. At present, only logs in the range of [-7x24 hours, +15 minutes] of the current time can be written. If the time stamp of any log is not within this range, the entire request fails, and no logs are written.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message	Description
400	PostBodyInvalid	Protobuffer content cannot be parsed.	Protobuffer content cannot be parsed.
400	InvalidTimestamp	Invalid time stamps are in logs.	Invalid time stamps are found in logs.
400	InvalidEncoding	Non-UTF8 characters are in logs.	Non-UTF8 characters are found in logs.
400	InvalidKey	Invalid keys are in logs.	Invalid keys are found in logs.
400	PostBodyTooLarge	Logs must be less than 3M and 4096 lines.	Logs must be less than 3 MB and 4,096 lines.
400	PostBodyUncompressError	Body is uncompressed fail.	Decompressing logs failed.
499	PostBodyInvalid	The post data time is out of range	The log time is out of [-7 × 24 hours, +15 minutes].
404	LogStoreNotExist	logstore {Name} not exist.	The LogStore does not exist.

The {name} variable in the above error messages will be replaced by the actual logstore name.

Example

Request example:

```
POST /logstores/sls-test-logstore
{
"Content-Length": 118,
"Content-Type":"application/x-protobuf",
"x-log-bodyrawsize":1356,
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Content-MD5":"6554BD042149C844761C2C094A8FECCE",
"Date":"Thu, 12 Nov 2015 06:54:26 GMT",
"x-log-apiversion": "0.6.0",
"x-log-compresstype":"lz4"
"x-log-signaturemethod": "hmac-sha1",
"Authorization":"LOG 94to3z418yupi6ikawqqd370:zLyKtgyGpwyv7ntXZs2dY2wWIg4="
}

<Binary data from logs in PB format compressed with LZ4>
```

Response example:

```
Header
{
"date": "Thu, 12 Nov 2015 06:53:03 GMT",
"connection": "close",
"x-log-requestid": "5644160399248C060600D216",
"content-length": "0",
"server": "nginx/1.6.1"
}
```

GetShipperStatus

Queries the log posting task status.

Request syntax

```
GET
/logstores/{logstoreName}/shipper/{shipperName}/tasks?from=1448748198&to=1448948198&status=success&off
set=0&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The LogStore name, unique under a project
shipperName	string	Yes	The name of the log posting rule, unique under a LogStore

from	integer	Yes	The time range for log posting task creation
to	integer	Yes	The time range for log posting task creation
status	string	No	The default value is empty, indicating that tasks in all statuses are returned. Currently, tasks in success, fail, or running status are supported.
offset	integer	No	The start number of posting tasks returned to a specified time range, 0 by default
size	integer	No	The start number of posting tasks returned to a specified time range, 0 by default

Request header

The GetShipperStatus interface does not have a special request header. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

The GetShipperStatus interface does not have a special response header. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

When a request is successful, the response body will contain a list of specified log posting tasks.

```
{
"count" : 10,
"total" : 20,
"statistics" : {
"running" : 0,
"success" : 20,
"fail" : 0
}
"tasks" : [
{
"id" : "abcdefghijk",
"taskStatus" : "success",
"taskMessage" : "",
"taskCreateTime" : 1448925013,
"taskLastDataReceiveTime" : 1448915013,
"taskFinishTime" : 1448926013
}
]
}
```

Name	Type	Description
count	integer	The number of returned tasks.
total	intege	The total number of tasks within a specified range.
statistics	json	Indicates the status of task summary within a specified range. For details, refer to the table below.
tasks	array	Indicates details about posting tasks within a specified range, as shown in the table below.

Statistics content:

Name	Type	Description
running	integer	The number of tasks in running status within a specified range.
success	integer	The number of tasks in success status within a specified range.
fail	integer	The number of tasks in fail status within a specified range.

Tasks content:

Name	Type	Description
id	string	The unique ID of a posting task.
taskStatus	string	The status of a posting task, which may be running, success, or fail.
taskMessage	string	The error message when the posting task fails.
taskCreateTime	integer	The posting task creation

		time.
taskLastDataReceiveTime	integer	The time of last log data in a posting task.
taskFinishTime	integer	The finish time of a posting task.

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	Project {ProjectName} not exist
404	LogStoreNotExist	logstore {logstoreName} not exist
400	ShipperNotExist	shipper {logstoreName} not exist
500	InternalServerError	internal server error
400	ParameterInvalid	start time must litter than end time
400	ParameterInvalid	only support query last 48 hours task status
400	ParameterInvalid	status only contains success/running/fail

Detailed description

The time range in which the posting task status can be queried must be within the last 24 hours.

Request example:

```
GET /logstores/test-logstore/shipper/test-
shipper/tasks?from=1448748198&to=1448948198&status=success&offset=0&size=100 HTTP/1.1
Header:
{
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:wFcl3ohVJupCi0ZFxRD0x4IA68A=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 08:28:19 GMT,
Content-Length=55,
x-log-signaturemethod=hmac-sha1,
Content-MD5=757C60FC41CC7D3F60B88E0D916D051E,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
```

Response example:

```
HTTP/1.1 200 OK
Header:
{
Date=Wed, 11 Nov 2015 08:28:20 GMT,
Content-Length=0,
x-log-requestid=5642FC2399248C8F7B0145FD,
Connection=close,
Server=nginx/1.6.1
}
Body:
{
"count" : 10,
"total" : 20,
"statistics" : {
"running" : 0,
"success" : 20,
"fail" : 0
}
"tasks" : [
{
"id" : "abcdefghijk",
"taskStatus" : "success",
"taskMessage" : "",
"taskCreateTime" : 1448925013,
"taskLastDataReceiveTime" : 1448915013,
"taskFinishTime" : 1448926013
}
]
}
```

RetryShipperTask

Re-executes failed log posting tasks.

Request syntax

```
PUT /logstores/{logstoreName}/shipper/{shipperName}/tasks HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

["task-id-1", "task-id-2", "task-id-2"]
```

Request parameters

Parameter Name	Type	Required or Not	Description

logstoreName	string	Yes	The LogStore name, unique under a project
shipperName	string	Yes	The name of the log posting rule, unique under a LogStore

Request header

The RetryShipperTask interface does not have a special request header. For details about public request headers of Log Service APIs, refer to Public Request Headers.

Response header

The RetryShipperTask interface does not have a special response header. For details about public response headers of Log Service APIs, refer to Public Response Headers.

Response element

The system returns the HTTP status code 200.

Error code

The CreateLogstore interface may return the following specific error codes in addition to Log Service API Common Error Codes:

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	Project {ProjectName} not exist
404	LogStoreNotExist	logstore {logstoreName} not exist
400	ShipperNotExist	shipper {logstoreName} not exist
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	only allow retry 10 task every time

Detailed description

A maximum of 10 failed posting tasks can be re-executed at a time.

Example

Request example:

```
PUT /logstores/test-logstore/shipper/test-shipper/tasks HTTP/1.1
Header:
{
x-log-apiversion=0.6.0,
Authorization=LOG 94to3z418yupi6ikawqqd370:wFcl3ohVJupCi0ZFxRD0x4IA68A=,
Host=ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com,
Date=Wed, 11 Nov 2015 08:28:19 GMT,
Content-Length=55,
x-log-signaturemethod=hmac-sha1,
Content-MD5=757C60FC41CC7D3F60B88E0D916D051E,
User-Agent=sls-java-sdk-v-0.6.0,
Content-Type=application/json
}
Body :
["task-id-1", "task-id-2", "task-id-2"]
```

Response example:

```
HTTP/1.1 200 OK
Header:
{
Date=Wed, 11 Nov 2015 08:28:20 GMT,
Content-Length=0,
x-log-requestid=5642FC2399248C8F7B0145FD,
Connection=close,
Server=nginx/1.6.1
}
```

GetLogs

On the GetHistograms interface, queries the log data in a LogStore under a specified project. By specifying the relevant parameters, it can also just query log data matching the specified criteria.

When a log is written in a Logstore, the delay at which log query interfaces (GetHistograms and GetLogs) can query this log varies with the log type. Based on a log’ s time stamp, the log service classifies it as one of two types:

- Real-time data: The time point for a log is the current time point for the sever (-180 seconds, 900 seconds]. For example, if the log time is UTC 2014-09-25 12:03:00 and the time point at which the server receives the log is UTC 2014-09-25 12:05:00, the log is used as the real-time data, which usually appears in normal scenarios.
- Historical data: The time point for a log is the current time point for the sever (-7 x 86400 seconds, -180 seconds]. For example, if the log time is UTC 2014-09-25 12:00:00 and the time point at which the server receives the log is UTC 2014-09-25 12:05:00, the log is used as the real-time data, which usually appears in scenarios of supplementation.

The maximum delay between real-time data writing and query is 3 seconds. (99.9% of the time the data can be queried within 1 second).

Request syntax

```
GET
/logstores/<logstorename>?type=histogram&topic=<logtopic>&from=<starttime>&to=<endtime>&query=<qu
erystring>&line=<linenum>&offset=<startindex>&reverse=<ture|false> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-bodyrawsize: 0
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Name	Type	Required	Description
logstorename	String	Yes	The LogStore name of the logs to query.
type	String	Yes	The type of the LogStore data to query. On the GetLogs interface, this parameter must be “log”.
from	Integer	Yes	The query start time (to the second, from 1970-1-1 00:00:00 UTC).
to	Integer	Yes	The query end time (to the second, from 1970-1-1 00:00:00 UTC).
topic	String	No	The log topic to query.
query	String	No	Query expression. For details about the query expression syntax, refer to Query Syntax.
line	Integer	No	The maximum number of log lines returned for the request. The value ranges from 0 to 100. The default value is 100.
offset	Integer	No	The returned log

			start point for the request. The value can be 0 or a positive integer. The default value is 0.
reverse	Boolean	No	Whether or not log are returned in reverse order according to the log time stamp. “true” indicates reverse order and “false” indicates regular order. The default value is “false”.

Request header

The GetLogs interface does not have a special request header. For details about the public request header of the log service API, refer to Public Request Header.

Response header

The GetLogs interface does not have a special response header. For details about the public response header of the log service API, refer to Public Response Header.

Response element

When a GetLogs request is successful, the response body will contain log data hit in the query. When you need to query an extremely large amount of log data, the response results from this interface may be incomplete. Therefore, the response element contains a specialized member that indicates whether or not the request’ s return results are complete. The specific response element format is as follows:

Name	Type	Description
progress	String	The query results status. The two values, Incomplete and Complete, indicate whether or not the results are complete.
count	Integer	The total number of logs in the current query results.
logs	Array	The original log data for the current query results. The specific structure is described below.

The structure of each element in the logs array is shown below:

Name	Type	Description

__time__	Integer	The log time stamp (to the second, from 1970-1-1 00:00:00 UTC).
__source__	String	The log source, specified when logs are written.
[content]	Key-Value pair	Original log content, organized in Key-value pair form.

Detailed description

- All the time ranges used in this interface follow the "Left closed and right open" principle. That is, these time ranges include the start time, but not the end time. If the from and to values are the same, the time range is invalid and the function will return an error.
- As described above, each call to this interface must return results within a limited time range, and thus each query can only scan a specified number of logs. If a request requires that an extremely large amount of data be processed, the return results for the request may be incomplete (the progress member in the return results indicates whether or not they are complete). At the same time, the server will cache query results within 15 minutes. When a portion of the query request results are cache hits, the server will continue to scan the log data that are not cache hits for this request. To reduce the workload of merging multiple query results, the server cache hit query results and the new hits of the current query are merged and returned to you. Therefore, the log service allows you to call the interface multiple times with the same parameters to finally obtain the complete results. Because the log data volume involved in the query changes massively, the log service API cannot predict how many calls are required to obtain complete results from the interface. Therefore, you must check the progress member value in the return results of each request to determine if they need to continue. You must note that each duplicate call to this interface will consume the same number of query CUs again.

Specific error codes

In addition to general error codes of the log service API, the GetLogs interface may return the following special error codes:

HTTP Status Code	Error Code	Error Message	Description
404	LogStoreNotExist	logstore {Name} not exist.	The LogStore does not exist.
400	InvalidTimeRange	request time range is invalid.	The request time range is invalid.
400	InvalidQueryString	query string is invalided	The request's query string is invalid.
400	InvalidOffset	offset is invalided	The request's offset parameter is invalid.

400	InvalidLine	line is invalided	The request’ s line parameter is invalid.
400	InvalidReverse	Reverse value is invalid	The Reverse parameter value is invalid.

The {name} variable in the above error messages will be replaced by the actual logstore name.

Example

Take a project named big-game in Hangzhou as an example. Query the logs themed groupA in the LogStore named app_log. The query ranges from 2014-09-01 00:00:00 to 2014-09-01 22:00:00, and the keyword is “error” . The query starts from the range header, and a maximum of 20 logs are returned.

Request example:

```
GET
/logstores/app_log ? type=log&topic=groupA&from=1409529600&to=1409608800&query=error&line=20&offset
=0 HTTP/1.1
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.4.0
x-log-signaturemethod: hmac-sha1
```

Response example:

```
HTTP/1.1 200 OK
Content-MD5: 36F9F7F0339BEAF571581AF1B0AAAFB5
Content-Type: application/json
Content-Length: 269
Date: Wed, 3 Sept. 2014 08:33:47 GMT
x-log-requestid: efag01234-12341-15432f

{
"progress": "Complete",
"count": 2,
"logs": [
{
"__time__": 1409529660,
"__source__": "10.237.0.17",
"Key1": "error",
"Key2": "Value2"
},
{
"__time__": 1409529680,
"__source__": "10.237.0.18",
"Key3": "error",
"Key4": "Value4"
}
]
}
```

In this response example, the progress member status is Complete. This indicates that the entire LogStore has been queried and the return results are complete. For this request, a total of two logs meet the query criteria and the log data is in the “logs” member. If, in the response results, the progress member has a status of Incomplete, you need to repeat the request to obtain the complete results.

GetHistograms

On the GetHistograms interface, queries the log distribution in a Logstore under a specified project. By specifying the relevant parameters, it can also just query log distributions matching the specified criteria.

When a log is written in a Logstore, the delay at which log query interfaces (GetHistograms and GetLogs) can query this log varies with the log type. Based on a log’ s time stamp, the log service classifies it as one of two types:

- Real-time data: The time point for a log is the current time point for the sever (-180 seconds, 900 seconds]. For example, if the log time is UTC 2014-09-25 12:03:00 and the time point at which the server receives the log is UTC 2014-09-25 12:05:00, the log is used as the real-time data, which usually appears in normal scenarios.
- Historical data: The time point for a log is the current time point for the sever (-7 x 86400 seconds, -180 seconds]. For example, if the log time is UTC 2014-09-25 12:00:00 and the time point at which the server receives the log is UTC 2014-09-25 12:05:00, the log is used as the real-time data, which usually appears in scenarios of supplementation.

The delay between real-time data writing and query is 1 minute.

Request syntax

```
GET
/logstores/<logstorename>?type=histogram&topic=<logtopic>&from=<starttime>&to=<endtime>&query=<qu
erystring> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-bodyrawsize: 0
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Name	Type	Required	Description
logstorename	String	Yes	The LogStore name of the logs to query.
type	String	Yes	The type of LogStore data to query. In the GetHistograms interface, this parameter must be set to “histogram”.
from	Integer	Yes	The query start time (to the second, from 1970-1-1 00:00:00 UTC).
to	Integer	Yes	The query end time (to the second, from 1970-1-1 00:00:00 UTC).
topic	String	No	The log topic to query.
query	String	No	Query expression. For details about the query expression syntax, refer to Query Syntax.

Request header

The GetHistograms interface does not have a special request header. For details about the public request header of the log service API, refer to Public Request Header.

Response header

The GetHistograms interface does not have a special response header. For details about the public response header of the log service API, refer to Public Response Header.

Response element

When a GetHistograms request is successful, the response body will contain the distribution of the number of log hits returned by the query along the time axis. The response results will evenly divide the time range queried by you into several (1 to 60) subintervals and return the number of log hits for each subinterval. Because the SLS must return results within a limited time range to ensure real-time functionality, each query can only scan a specified number of logs. When you need to query an extremely large amount of log data, the response results from this interface may be incomplete.

Therefore, the response element contains a specialized member that indicates whether or not the request’ s return results are complete. The specific response element format is as follows:

Name	Type	Description
progress	String	The query results status. The two values, Incomplete and Complete, indicate whether or not the results are complete.
count	Integer	The total of all logs in the current query results.
histograms	Array	The distribution of the current query results among the subintervals. For the actual structure, see the description below.

The structure of each element in the histograms array is shown below:

Name	Type	Description
from	Integer	The start time for subintervals (to the second, from 1970-1-1 00:00:00 UTC
to	Integer	The end time for subintervals (to the second, from 1970-1-1 00:00:00 UTC)
count	Integer	The number of log hits for this subinterval in the query results
progress	String	Indicates whether or not the query results for this subinterval are complete. Values: Incomplete and Complete.

Detailed description

- All the time ranges used in this interface (whether it is a time range defined by a request’ s from and to parameters or the subintervals in the return results) follow the “Left closed and right open” principle. That is, these time ranges include the start time, but not the end time. If the from and to values are the same, the time range is invalid and the function will return an error.
- The subinterval division method for the responses from this interface are consistent and unchanging. If the time range queried by you does not change, neither will the subinterval division in the response.
- As described above, each call to this interface must return results within a limited time range,

and thus each query can only scan a specified number of logs. If a request requires that an extremely large amount of data be processed, the return results for the request may be incomplete (the progress member in the return results indicates whether or not they are complete). At the same time, the server will cache query results within 15 minutes. When a portion of the query request results are cache hits, the server will continue to scan the log data that are not cache hits for this request. To reduce the workload of merging multiple query results, the server cache hit query results and the new hits of the current query are merged and returned to you. Therefore, the log service allows you to call the interface multiple times with the same parameters to finally obtain the complete results. Because the log data volume involved in the query changes massively, the log service API cannot predict how many calls are required to obtain complete results from the interface. Therefore, you must check the progress member value in the return results of each request to determine if they need to continue. You must note that each duplicate call to this interface will consume the same number of query CUs again.

Specific error codes

In addition to general error codes of the log service API, the GetHistograms interface may return the following special error codes:

HTTP Status Code	Error Code	Error Message	Description
404	LogStoreNotExist	logstore {Name} not exist.	The LogStore does not exist.
400	InvalidTimeRange	request time range is invalid.	The request time range is invalid.
400	InvalidQueryString	query string is invalided	The request’s query string is invalid.

The {name} variable in the above error messages will be replaced by the actual logstore name.

Example

Take a project named big-game in Hangzhou as an example. Query the distribution of logs themed groupA in the LogStore named app_log. The query ranges from 2014-09-01 00:00:00 to 2014-09-01 22:00:00, and the keyword is “error”.

Request example:

```
GET /logstores/app_log?type=histogram&topic=groupA&from=1409529600&to=1409608800&query=error
HTTP/1.1
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
```

```
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Response example:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-MD5: E6AD9C21204868C2DE84EE3808AAA8C8
Content-Type: application/json
Date: Wed, 3 Sept. 2014 08:33:47 GMT
Content-Length: 232
x-log-requestid: efag01234-12341-15432f

{
"progress": "Incomplete",
"count": 3,
"histograms": [
{
"from": 1409529600,
"to": 1409569200,
"count": 2,
"progress": "Complete"
},
{
"from": 1409569200,
"to": 1409608800,
"count": 1,
"progress": "Incomplete"
}
]
}
```

In this response example, the server divides the entire Histogram into two equal time ranges: [2014-09-01 00:00:00, 2014-09-01 11:00:00) and [2014-09-01 11:00:00, 2014-09-01 22:00:00). Because the amount of your data is too large, the first query will return incomplete data. The response results indicate that the number of log hits is 3, but the overall results are incomplete. Only the results for the time range [2014-09-01 00:00:00 2014-09-01 11:00:00) are complete with 2 hits, while the results for the other time range are incomplete with 1 hit. In this situation, if you wish to obtain the complete results, you must duplicate the above call request several times until the “progress” member for the overall results changes to “Complete” (as shown in the response below):

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-MD5: E6AD9C21204868C2DE84EE3808AAA8C8
Content-Type: application/json
Date: Wed, 3 Sept. 2014 08:33:48 GMT
Content-Length: 232
x-log-requestid: afag01322-1e241-25432e

{
"progress": "Incomplete",
"count": 4,
"histograms": [
{
"from": 1409529600,
"to": 1409569200,
"count": 2,
"progress": "Complete"
},
{
"from": 1409569200,
"to": 1409608800,
"count": 2,
"progress": "complete"
}
]
}
```

Logtail machine group related interfaces

CreateMachineGroup

You can create a group of machines to collect logs and deliver configuration.

Example:

```
POST /machinegroups
```

Request syntax

```
POST /machinegroups HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1


{
"groupName" : "testgroup",
"groupType" : "",
"groupAttribute" : {
```

```
"externalName" : "testgroup",
"groupTopic": "testgrouptopic"
},
"machineIdentifyType" : "ip",
"machineList" : [
"test-ip1",
"test-ip2"
]

}
```

Request parameters

Body parameters:

Name	Type	Required or Not	Description
groupName	string	Yes	The machine group name, which is unique under a project
groupType	string	No	The machine group type, which is empty by default
machineIdentifyType	string	Yes	The machine identifier type, including IP and userdefined
groupAttribute	object	Yes	The machine group attribute, which is empty by default
machineList	array	Yes	The specific machine ID, which can be an IP address or a user-defined ID

Group attributes are described as follows:

Name	Type	Required or Not	Description
groupTopic	string	No	The topic of a machine group, which is empty by default
externalName	string	No	The external management ID, which is empty by default

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

I addition to general error codes of the log service API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
400	GroupAlreadyExist	group {GroupName} already exist
400	InvalidParameter	invalid group resource json
500	InternalServerError	Internal server error

Detailed description

None

Example

Request example:

```
POST /machinegroups HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:aws39CB5OUyx39BjQ5bW3G/zBv4=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 17:57:33 GMT",
"Content-Length": "187",
"x-log-signaturemethod": "hmac-sha1",
"Content-MD5": "82033D507DEAAD72067BB58DFDCB590D",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/json",
"x-log-bodyrawsize": "0"
}
Body :
{
"groupName": "test-machine-group",
"groupType": "",
"machineIdentifyType": "ip",
"groupAttribute": {
"groupTopic": "testtopic",
"externalName": "testgroup"
},
"machineList": [
"127.0.0.1",
"127.0.0.2"
]
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 17:57:33 GMT",
"Content-Length": "0",
"x-log-requestid": "5642300D99248CB76D005D36",
"Connection": "close",
"Server": "nginx/1.6.1"
}
```

DeleteMachineGroup

Deletes the machine group. If the machine group is configured, the Logtail configuration will be deleted.

Example:

```
DELETE /machinegroups/{groupName}
```

Request syntax

```
DELETE /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description

groupName	string	Yes	The machine group name

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

HTTP Status Code	ErrorCode	Error Message
404	GroupNotExist	group {GroupName} not exist
500	InternalServerError	internal server error

Detailed description

None

Example

Request example:

```
DELETE /machinegroups/test-machine-group-4 HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:JjQpxvfnkTYPsZIgicQ+IOkufI8=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 19:13:28 GMT",
"Content-Length": "0",
"x-log-signaturemethod": "hmac-sha1",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/x-protobuf",
"x-log-bodyrawsize": "0"
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 19:13:28 GMT",
"Content-Length": "0",
"x-log-requestid": "564241D899248C827B000CFE",
"Connection": "close",
"Server": "nginx/1.6.1"
}
```

UpdateMachineGroup

Updates the machine group information. If a configuration is already applied to a machine group, the configuration is automatically applied to or removed from the machines added to or deleted from the machine group.

Example:

```
PUT /machinegroups/{groupName}
```

```
PUT /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

{
"groupName": "test-machine-group",
"groupType" : "",
"groupAttribute" : {
"externalName" : "testgroup",
"groupTopic": "testgrouptopic"
},
"machineIdentifyType" : "ip",
"machineList" : [
"test-ip1",
"test-ip2"
]
}
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description
groupName	string	Yes	The machine group name

Body parameters:

Name	Type	Required or Not	Description
groupName	string	Yes	The machine group name, which is unique under a project
groupType	string	No	The machine group type, which is empty by default
machineIdentifyType	string	Yes	The machine identifier type, including IP and userdefined
groupAttribute	object	Yes	The machine group attribute, which is empty by default
machineList	array	Yes	The specific machine ID, which can be an IP address or a user-defined ID

Group attributes are described as follows:

Name	Type	Required or Not	Description
groupTopic	string	No	The topic of a machine group, which is empty by default
externalName	string	No	The external management ID, which is empty by default

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

I addition to general error codes of the log service API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	GroupNotExist	group {GroupName} not exist
400	InvalidParameter	invalid group resource json
500	InternalServerError	internal server error

Detailed description

None

Example

Request example:

```
PUT /machinegroups/test-machine-group HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:ZJmBDS+LjRCzgSLuo21vFh6o7CE=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 18:41:43 GMT",
"Content-Length": "194",
"x-log-signaturemethod": "hmac-sha1",
"Content-MD5": "2CEBAEBE53C078891527CB70A855BAF4",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/json",
"x-log-bodyrawsize": "0"
}
Body :
{
"groupName": "test-machine-group",
"groupType": "",
"machineIdentifyType": "userdefined",
"groupAttribute": {
"groupTopic": "testtopic2",
"externalName": "testgroup2"
},
"machineList": [
"uu_id_1",
"uu_id_2"
]
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 18:41:43 GMT",
"Content-Length": "0",
"x-log-requestid": "56423A6799248CA57B00035C",
"Connection": "close",
"Server": "nginx/1.6.1"
}
```

ListMachineGroup

Example:

GET /machinegroups?offset=1&size=100

```
GET /machinegroups?offset=1&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description
offset	int	No	Start position of the returned results. The default value is 0.
size	int	No	Maximum number of entries returned per page. The default value is 500 (maximum).
groupName	string	No	Group machine name used for filtering (partial

			matching supported)

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

When the request is successful, the response Body contains a list of all machine groups in the specified project. The specific format is as follows:

Name	Type	Description
count	int	The number of returned machine groups
total	int	Total number of returned machine groups
machinegroups	json array	List of returned machine groups

```
{
"machinegroups": [
"test-machine-group",
"test-machine-group-2"
],
"count": 2,
"total": 2
}
```

Error code

I addition to general error codes of the log service API, the following special error codes may be returned:

HTTP status code	Error code	Error message
500	InternalServerError	internal server error

Detailed description

None

Example

Request example:

```
GET /machinegroups?groupName=test-machine-group&offset=0&size=3 HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:XN5helROz9QYV0FKhElSNuTBysA=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 18:34:44 GMT",
"Content-Length": "0",
"x-log-signaturemethod": "hmac-sha1",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/x-protobuf",
"x-log-bodyrawsize": "0"
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 18:34:44 GMT",
"Content-Length": "83",
"x-log-requestid": "564238C499248C8F7B0001DE",
"Connection": "close",
"Content-Type": "application/json",
"Server": "nginx/1.6.1"
}
Body :
{
"machinegroups": [
"test-machine-group",
"test-machine-group-2"
],
"count": 2,
"total": 2
}
```

GetMachineGroup

Gets details of a machine group.

Example:

GET /machinegroups/{groupName}

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description
groupName	string	Yes	The machine group name

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

Name	Type	Description	
groupName	string	The machine group name, which is unique under a project	
groupType	string	Group type (null or Armory). The default value is null.	
machineIdentifyType	string	Type of machine ID, which can be an IP address or userdefined.	
groupAttribute	json object	Attribute of the machine group. The default value is empty.	
machineList	json array	The specific machine ID, which can be an	

		IP address or a user-defined ID	
createTime	int	Creation time of the machine group	
lastModifyTime	int	Last update time of the machine group	

Group attributes are described as follows:

Name	Type	Required or Not	Description
groupTopic	string	No	Topic of the machine group, which is empty by default
externalName	string	No	ID of the external management system (Armory) of the machine group

```
{
"groupName": "test-machine-group",
"groupType": "",
"groupAttribute": {
"externalName": "testgroup",
"groupTopic": "testtopic"
},
"machineIdentifyType": "ip",
"machineList": [
"127.0.0.1",
"127.0.0.2"
],
"createTime": 1447178253,
"lastModifyTime": 1447178253
}
```

Error code

I addition to general error codes of the log service API, the following special error codes may be returned:

HTTP Status Code	ErrorCode	Error Message
404	GroupNotExist	group {GroupName} not exist
500	InternalServerError	internal server error

Detailed description

None

Example

Request example:

```
GET /machinegroups/test-machine-group HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:CNQaXNeExV6S/nQZkP/R+baZPZc=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 18:15:24 GMT",
"Content-Length": "0",
"x-log-signaturemethod": "hmac-sha1",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/x-protobuf",
"x-log-bodyrawsize": "0"
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 18:15:23 GMT",
"Content-Length": "239",
"x-log-requestid": "5642343B99248CB36D0060B8",
"Connection": "close",
"Content-Type": "application/json",
"Server": "nginx/1.6.1"
}
Body :
{
"groupName": "test-machine-group",
"groupType": "",
"groupAttribute": {
"externalName": "testgroup",
"groupTopic": "testtopic"
},
"machineIdentifyType": "ip",
"machineList": [
"127.0.0.1",
"127.0.0.2"
],
"createTime": 1447178253,
"lastModifyTime": 1447178253
}
```

ApplyConfigToMachineGroup

Applies configuration to the machine group.

Example:

```
PUT /machinegroups/{GroupName}/configs/{ConfigName}
```

Request parameters

Name	Type	Required or Not	Description
groupName	string	Yes	The machine group name
ConfigName	string	Yes	Name of the log configuration

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	GroupNotExist	group {GroupName} not exist
404	ConfigNotExist	config {ConfigName} not exist
500	InternalServerError	internal server error

Example

Request example:

```
PUT /machinegroups/sample-group/configs/logtail-config-sample

Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Mon, 09 Nov 2015 09:44:43 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:skTdJCZXn8QPGNB2jL9k6u1xO1E="
}
```

Response example:

```
{
"date": "Mon, 09 Nov 2015 09:44:43 GMT",
"connection": "close",
"x-log-requestid": "56406B0B99248CAA230BA094",
"content-length": "0",
"server": "nginx/1.6.1"
}
```

RemoveConfigFromMachineGroup

Deletes configurations from the machine group.

Example:

```
DELETE /machinegroups/{GroupName}/configs/{ConfigName}
```

Request parameters

Name	Type	Required or Not	Description
GroupName	string	Yes	Name of the machine group
ConfigName	string	Yes	Name of the log configuration

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	GroupNotExist	group {GroupName} not exist
404	ConfigNotExist	config {ConfigName} not exist
500	InternalServerError	internal server error

Example

Request example:

```
DELETE /machinegroups/sample-group/configs/logtail-config-sample

{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Mon, 09 Nov 2015 09:48:48 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:t8v8y+zqOz3ZiqLDIkb6JQ8FUAU="
}
```

Response example:

```
{
"date": "Mon, 09 Nov 2015 09:48:48 GMT",
"connection": "close",
"x-log-requestid": "56406C0099248CAA230BE135",
"content-length": "0",
"server": "nginx/1.6.1"
}
```

ListMachines

Gets the status of your server-connected machines in the specified machine group.

Example:

```
GET /machinegroups/{groupName}/machines?offset=1&size=10
```

```
GET /machinegroups/{groupName}/machines HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required	Description
groupName	string	Yes	Name of the machine group
offset	int	No	Start position of the returned results. The default value is 0.
size	int	No	Maximum number of entries returned per page. The default value is 500 (maximum).

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

Name	Type	Description

count	int	The number of returned machines
total	int	Total number of machines
machines	json array	List of returned machines

Machines are described as follows:

Name	Type	Description
ip	string	Machine IP
machine-uniqueid	string	Machine DMI UUID
userdefined-id	string	Custom ID of the machine

```
{
"count":10,
"total":100,
"machines":
[{
"ip" : "testip1",
"machine-uniqueid" : "testuuid1",
"userdefined-id" : "testuserdefinedid1",
"lastHeartbeatTime" : 1447182247
},
{
"ip" : "testip1",
"machine-uniqueid" : "testuuid2",
"userdefined-id" : "testuserdefinedid2",
"lastHeartbeatTime" : 1447182247
},
{
"ip" : "testip2",
"machine-uniqueid" : "testuuid",
"userdefined-id" : "testuserdefinedid"
"lastHeartbeatTime" : 1447182247
}]
}
```

Error code

In addition to general error codes of the log service API, the following special error codes may be returned:

HTTP Status Code	ErrorCode	Error Message
404	GroupNotExist	group {GroupName} not exist
500	InternalServerError	internal server error

Detailed description

Only the list of machines in normal connection with the server can be obtained at this interface.

Example

Request example:

```
GET /machinegroups/test-machine-group-5/machines?offset=0&size=3 HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:9yoK0iJPxr0RrWf/wW9NJYXu4zo=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 19:04:57 GMT",
"Content-Length": "0",
"x-log-signaturemethod": "hmac-sha1",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/x-protobuf",
"x-log-bodyrawsize": "0"
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 19:04:58 GMT",
"Content-Length": "324",
"x-log-requestid": "56423FD999248C827B000A57",
"Connection": "close",
"Content-Type": "application/json",
"Server": "nginx/1.6.1"
}
Body :
{
"machines": [
{
"ip": "10.101.166.116",
"machine-uniqueid": "",
"userdefined-id": "",
"lastHeartbeatTime": 1447182247
},
{
"ip": "10.101.165.193",
"machine-uniqueid": "",
"userdefined-id": "",
"lastHeartbeatTime": 1447182246
},
{
"ip": "10.101.166.91",
"machine-uniqueid": "",
"userdefined-id": "",
"lastHeartbeatTime": 1447182248
}
],
"count": 3,
"total": 8
}
```

GetAppliedConfigs

Gets configurations applied in the machine group.

Example:

```
GET /machinegroups/{groupName}/configs
```

```
GET /machinegroups/{groupName}/configs HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description
groupName	string	Yes	Name of the machine group

Request header

No special request header is available. For details about the public request header of the log service API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the log service API, refer to Public Response Header.

Response element

When the request is successful. The response Body includes a list of all machines in the specified machine group in the following format:

Name	Type	Description
count	Integer	Number of returned configurations.
configs	String array	List of returned configuration names.

```
{
"count":2,
"configs":
["config1","config2"]
}
```

Error code

I addition to general error codes of the log service API, the following special error codes may be returned:

HTTP Status Code	ErrorCode	Error Message
404	GroupNotExist	group {GroupName} not exist
500	InternalServerError	internal server error

Detailed description

None

Example

Request example:

```
GET /machinegroups/test-machine-group/configs HTTP/1.1
Header :
{
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:/Ntg290OaJ8JfInmhzYTG/GJwbE=",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Tue, 10 Nov 2015 19:45:48 GMT",
"Content-Length": "0",
"x-log-signaturemethod": "hmac-sha1",
"User-Agent": "sls-java-sdk-v-0.6.0",
"Content-Type": "application/x-protobuf",
"x-log-bodyrawsize": "0"
}
```

Response example:

```
HTTP/1.1 200 OK
Header :
{
"Date": "Tue, 10 Nov 2015 19:45:48 GMT",
"Content-Length": "53",
"x-log-requestid": "5642496C99248C8C7B00173F",
"Connection": "close",
"Content-Type": "application/json",
"Server": "nginx/1.6.1"
}
Body :
{
"configs": [
"two",
"three",
"test_logstore"
],
"count": 3
}
```

Logtail configuration related interfaces

CreateConfig

Creates log configuration under a project.

Example:

    POST /configs

Request syntax

```
POST /configs HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

{
```

```
"configName": "testcategory1",
"inputType": "file",
"inputDetail": {
"logType": "common_reg_log",
"logPath": "/var/log/httpd/",
"filePattern": "access*.log",
"localStorage": true,
"timeFormat": "%Y/%m/%d %H:%M:%S",
"logBeginRegex": ".*",
"regex": "(\w+)(\s+)",
"key" :["key1", "key2"],
"filterKey":["key1"],
"filterRegex":["regex1"],
"fileEncoding":"utf8",
"topicFormat": "none"
},
"outputType": "LogService",
"outputDetail":
{
"logstoreName": "perfcounter"
}
}
```

Request parameters

Attribute Name	Type	Required or Not	Description
configName	string	Yes	The log config name, which is unique in the project.
inputType	string	Yes	The input type, which must be file
inputDetail	json	Yes	See description in the following table.
outputType	string	Yes	The output type, which must be LogService
outputDetail	json	Yes	See the descriptions in the following table.
logSample	string	No	The Logtail log configuration sample. The log size must not exceed 1,000 bytes.

Content of inputDetail:

Attribute Name	Type	Required or Not	Description
logType	string	Yes	The log type. Currently, only common_reg_log is supported.
logPath	string	Yes	The parent directory where the log is located, for example, /var/logs/.
filePattern	string	Yes	The pattern of a log file, for example, access*.log
localStorage	boolean	Yes	Whether to activate local cache. Logs of 1 GB can be stored locally when the links across the servers are disconnected.
timeFormat	string	Yes	The format of log time, for example, %Y/%m/%d %H:%M:%S.
logBeginRegex	string	Yes	The characteristics (regular expression) of the first log line, used for a log composed of multiple lines
regex	string	Yes	The regular expression used for extracting a log pair.
key	array	Yes	The key generated after the log is extracted.
filterKey	array	Yes	The key used for filtering the log. The log meets requirements only when the key value matches the regular expression specified in the corresponding filterRegex column.
filterRegex	array	Yes	The regular expression corresponding to each filterKey. The length of filterRegex

			must be same as that of filterKey.
topicFormat	string	No	The topic generation mode. Four formats are supported: 1) A part of the log file path is used as the topic, for example, /var/log/(.*).log; 2) none, indicating that the topic is empty; 3) default, indicating that the log file path is used as a topic; 4) group_topic, indicating that the topic attribute of the configured machine group is used as a topic.
preserve	boolean	No	“true” indicates that the monitored directories will never be timed out and “false” indicates that the monitored directories have been timed out by 30 minutes. The default value is true.
preserveDepth	integer	No	If preserve is set to false, the depth of the directories with no monitoring timeout is specified. The maximum depth is 3.
fileEncoding	string	No	Two types are supported: utf8、gbk

Content of outputDetail:

Name	**Type**	**Required**	**Description**
logstoreName	string	Yes	The LogStore name

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
400	ConfigAlreadyExist	config {Configname} already exist
400	InvalidParameter	invalid config resource json
500	InternalServerError	internal server error

Detailed description

In case of existing configuration, format error, loss of necessary parameters, or reached quota, creation will fail.

Example

Request example:

```
POST /configs HTTP/1.1
Header :
{
'Content-Length': 737,
'Host': 'ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com',
'x-log-bodyrawsize': 737,
'Content-MD5': 'FBA01ECF7255BE143379BC70C56BBF68',
'x-log-signaturemethod': 'hmac-sha1',
'Date': 'Mon, 09 Nov 2015 07:45:30 GMT',
'x-log-apiversion': '0.6.0',
'User-Agent': 'log-python-sdk-v-0.6.0',
'Content-Type': 'application/json',
'Authorization': 'LOG 94to3z418yupi6ikawqqd370:x/L1ymdn9wxe2zrwzcdSG82nXL0='
}
Body:
{
"configName": "sample-logtail-config",
"inputType": "file",
"inputDetail": {
"logType": "common_reg_log",
"logPath": "/var/log/httpd/",
"filePattern": "access*.log",
"localStorage": true,
"timeFormat": "%d/%b/%Y:%H:%M:%S",
"logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
"regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+) ([^\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\"
\"([^\"]*)\".*",
"key": ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "browser"],
"filterKey": [],
"filterRegex": [],
"topicFormat": "none",
"fileEncoding": "utf8"
},
"outputType": "LogService",
"outputDetail":
{
"logstoreName": "sls-test-logstore"
}
}
```

Response example:

```
HTTP/1.1 200 OK
Header
{
'date': 'Mon, 09 Nov 2015 07:45:30 GMT',
'connection': 'close',
'x-log-requestid': '56404F1A99248CA26C002180',
'content-length': '0',
'server': 'nginx/1.6.1'
}
```

ListConfig

Lists all configuration information under a project. You can use the parameter to flip the page.

Example:

```
GET /configs?offset=1&size=100
```

Request syntax

```
GET /configs?offset=0&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters are as follows:

Name	Type	Required or Not	Description
offset (optional)	integer	No	The starting position of a returned record, 0 by default.
size (optional)	integer	No	The maximum number of entries returned per page, 500 by default (maximum).

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

Return values: Body contains a list of all configurations under this project. The format is as follows:

Name	Type	Description
count	Integer	The number of returned configurations.
total	Integer	The total number of configurations on the server
configs	String array	The list of returned configurations.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	ConfigNotExist	config {Configname} not exist
500	InternalServerError	internal server error

Detailed description

N/A

Example

Request example:

```
GET /configs?offset=0&size=10 HTTP/1.1

Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Mon, 09 Nov 2015 09:19:13 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:teWnMylnM4Toohhp9dfBECrEgac="
}
```

Response example:

```
Header :
{
"content-length": "103",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Mon, 09 Nov 2015 09:19:13 GMT",
"content-type": "application/json",
"x-log-requestid": "5640651199248CAA2300C2BA"
}

Body:
{
"count": 3,
"configs":
[
"logtail-config-sample",
"logtail-config-sample-2",
"logtail-config-sample-3"
],
"total": 3
}
```

GetAppliedMachineGroups

Lists configured machines.

Example:

```
GET /configs/{configName}/machinegroups
```

Request syntax

```
GET /configs/{configName}/machinegroups HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description
ConfigName	string	Yes	The configuration name

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

When the request is successful. The response Body includes a list of all machines in the specified machine group in the following format:

Name	Type	Description
count	Integer	The number of returned machine groups.
machinegroups	String array	The list of returned machine groups.

```
{
"count":2,
"machinegroups":
```

```
["group1","group2"]
}
```

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	ErrorCode	Error Message
404	GroupNotExist	group {GroupName} not exist
500	InternalServerError	internal server error

Detailed description

None

Example

Request example:

```
GET /configs/logtail-config-sample/machinegroups
Header:
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Mon, 09 Nov 2015 09:51:38 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:+6bo4MSUt/dyNa72kXeGCkVOi+4="
}
```

Response example:

```
Header :
{
"content-length": "44",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Mon, 09 Nov 2015 09:51:38 GMT",
"content-type": "application/json",
"x-log-requestid": "56406CAA99248CAA230BE828"
}

Body:
{
"count": 1,
"machinegroups":
```

```
[
"sample-group"
]
}
```

GetConfig

Obtains configuration details.

Example:

```
GET /configs/{configName}
```

Request syntax

```
GET /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

Name	Type	Required or Not	Description
ConfigName	string	Yes	The log configuration name

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

Name	Type	Description	
configName	string	The log configuration name, which is unique	

		under a project	
inputType	string	The input type, which must be file	
inputDetail	json	See the descriptions in the following table.	
outputType	string	The output type, which must be LogService	
outputDetail	json	See the descriptions in the following table.	
createTime	Int	The configuration creation time	
lastModifyTime	Int	The resource server update time	

Content of inputDetail:

Name	Type	Description	
logType	string	The log type, which must be common_reg_log	
logPath	string	The parent directory where a log resides, for example, /var/logs/	
filePattern	string	The pattern of a log file, for example, access*.log	
localStorage	boolean	Indicates whether to activate local cache so that 1 GB logs can be stored locally when the links across the servers are disconnected.	
timeFormat	string	The log time format, for example, %Y/%m/%d %H:%M:%S	
logBeginRegex	string	The characteristics (regular expression) of the first log line, used for a log composed of multiple lines	

regex	string	The regular expression used for extracting a log pair	
key	array	The key generated after a log is extracted	
filterKey	array	The key used for filtering a log. The log meets requirements only when the value of value matches the regular expression specified in the corresponding filterRegex column.	
filterRegex	array	The regular expression corresponding to each filterKey. The length of filterRegex must be same as that of filterKey.	
topicFormat	string	A part of a log file path is used as a topic, for example, /var/log/(.*).log. By default, this parameter is set to none, indicating that the topic is empty.	
preserve	boolean	“true” indicates that the monitored directories will never time out and “false” indicates that the monitored directories timed out for 30 minutes. The default value is “true”.	
preserveDepth	integer	If you set preserve to false, the depth of the directories with no monitoring timeout is specified. The maximum depth is 3.	

Content of outputDetail:

Name	Type	Required	Description

endpoint	string	No	The project address without a prefix "project". If you leave it blank, the default value is EndPoint.
logstoreName	string	Yes	The LogStore name

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	ConfigNotExist	Config {Configname} not exist
500	InternalServerError	Specified Server Error Message

Detailed description

N/A

Example

Request example:

```
GET /configs/logtail-config-sample
Header :
{
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Mon, 09 Nov 2015 08:29:15 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:yV5LsYLmn1UrAXvBg8CbZNZoiTk="
}
```

Response example:

```
Header :
{
"content-length": "730",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Mon, 09 Nov 2015 08:29:15 GMT",
"content-type": "application/json",
"x-log-requestid": "5640595B99248CAA23004A59"
}
Body :
{
"configName": "logtail-config-sample",
"outputDetail": {
"endpoint": "http://cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"logstoreName": "sls-test-logstore"
},
"outputType": "LogService",
"inputType": "file",
"inputDetail": {
"regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+) ([^\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\"
\"([^\"]*)\".*",
"filterKey": [],
"logPath": "/var/log/httpd/",
"logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
"logType": "common_reg_log",
"topicFormat": "none",
"localStorage": true,
"key": [
"ip",
"time",
"method",
"url",
"request_time",
"request_length",
"status",
"length",
"ref_url",
"browser"
],
"filePattern": "access*.log",
"timeFormat": "%d/%b/%Y:%H:%M:%S",
"filterRegex": []
},
"createTime": 1447040456,
"lastModifyTime": 1447050456
}
```

DeleteConfig

Deletes specific configuration. If the configuration has been applied to the machine group, the Logtail configuration will be deleted.

```
DELETE /configs/{configName}
```

Request syntax

```
DELETE /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

Request parameters

URL parameters:

Name	Type	Required or Not	Description
ConfigName	string	Yes	The log configuration name

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

The system returns the HTTP status code 200.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
400	ConfigAlreadyExist	config {Configname} already exist
400	InvalidParameter	invalid config resource json
500	InternalServerError	internal server error

Example

Request example:

```
DELETE /configs/logtail-config-sample
Header :
{
```

```
"Content-Length": 0,
"x-log-signaturemethod": "hmac-sha1",
"x-log-bodyrawsize": 0,
"User-Agent": "log-python-sdk-v-0.6.0",
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"Date": "Mon, 09 Nov 2015 09:28:21 GMT",
"x-log-apiversion": "0.6.0",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:utd/O1JNCYvcRGiSXHsjKhTzJDI="
}
```

Response example:

```
Header :
{
"date": "Mon, 09 Nov 2015 09:28:21 GMT",
"connection": "close",
"x-log-requestid": "5640673599248CAA230836C6",
"content-length": "0",
"server": "nginx/1.6.1"
}
```

UpdateConfig

Updates the configuration. If the configuration is applied to the machine group, the corresponding machine will be updated.

Example:

```
PUT /configs/{configName}
```

Request syntax

```
PUT /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1

{
"configName": "testcategory1",
"inputType": "file",
"inputDetail": {
"logType": "common_reg_log",
"logPath": "/var/log/httpd/",
"filePattern": "access.log",
"localStorage": true,
"timeFormat": "%Y/%m/%d %H:%M:%S",
"logBeginReg": ".*",
"regex": "(\w+)(\s+)",
"key" :["key1", "key2"],
"filterKey":["key1"],
"filterRegex":["regex1"],
"topicFormat": "none"
},
"outputType": "LogService",
"outputDetail":
{
"logstoreName": "perfcounter"
}
}
```

Request parameters

Attribute Name	Type	Required or Not	Description
configName	string	Yes	The log configuration name, which is unique under a project
inputType	string	Yes	The input type, which must be file
inputDetail	json	Yes	See description in the following table.
outputType	string	Yes	The output type, which must be LogService
outputDetail	json	Yes	See the descriptions in the following table.

Content of inputDetail:

Attribute Name	Type	Required or Not	Description
logType	string	Yes	The log type. Currently, only common_reg_log is supported.
logPath	string	Yes	The parent directory where the log is

			located, for example, /var/logs/.
filePattern	string	Yes	The pattern of a log file, for example, access*.log
localStorage	boolean	Yes	Whether to activate local cache. Logs of 1 GB can be stored locally when the links across the servers are disconnected.
timeFormat	string	Yes	The log time format, for example, %Y/%m/%d %H:%M:%S
logBeginRegex	string	Yes	The characteristics (regular expression) of the first log line, used in the case of a log composed of multiple lines.
regex	string	Yes	The regular expression used for extracting a log pair.
key	array	Yes	The key generated after the log is extracted.
filterKey	array	Yes	The key used for filtering the log. The log meets requirements only when the key value matches the regular expression specified in the corresponding filterRegex column.
filterRegex	array	Yes	The regular expression corresponding to each filterKey. The length of filterRegex must be same as that of filterKey.
topicFormat	string	No	This is used to take one part of a log file path as a topic, for example, /var/log/(.*).log. The default value is none, indicating that

			the topic is null.
preserve	boolean	No	"true" indicates that the monitored directories will never be timed out and "false" indicates that the monitored directories have been timed out by 30 minutes. The default value is true.
preserveDepth	integer	No	If preserve is set to false, the depth of the directories with no monitoring timeout is specified. The maximum depth is 3.
fileEncoding	string	No	Two types are supported: utf8 and gbk. The default value is utf8.

Content of outputDetail:

Name	Type	Required	Description
logstoreName	string	Yes	The LogStore name

Request header

No special request header is available. For details about the public request header of the API, refer to Public Request Header.

Response header

No special response header is available. For details about the public response header of the API, refer to Public Response Header.

Response element

Return values: The status code 200 is returned.

Error code

In addition to general error codes of the API, the following special error codes may be returned:

HTTP Status Code	Error Code	Error Message
404	ConfigNotExist	config {Configname} not exist

400	InvalidParameter	invalid config resource json
400	BadRequest	config resource configname not match request
500	InternalServerError	internal server error

Detailed description

In case of format error, loss of necessary parameters, or reached quota, creation will fail.

Example

Request example:

```
PUT /configs/logtail-config-sample
Header :
{
"Content-Length": 737,
"Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.aliyuncs.com",
"x-log-bodyrawsize": 737,
"Content-MD5": "431263EB105D584A5555762A81E869C0",
"x-log-signaturemethod": "hmac-sha1",
"Date": "Mon, 09 Nov 2015 09:14:32 GMT",
"x-log-apiversion": "0.6.0",
"User-Agent": "log-python-sdk-v-0.6.0",
"Content-Type": "application/json",
"Authorization": "LOG 94to3z418yupi6ikawqqd370:GTPzFbLe8PZW0OFxTk/xMoCXA9E="
}
Body :
{
"outputDetail": {
"logstoreName": "sls-test-logstore"
},
"inputType": "file",
"inputDetail": {
"regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+) ([^\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\"
\"([^\"]*)\".*",
"filterKey": [],
"logPath": "/var/log/nginx/",
"logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
"logType": "common_reg_log",
"topicFormat": "none",
"localStorage": true,
"key": [
"ip",
"time",
"method",
"url",
"request_time",
"request_length",
"status",
"length",
"ref_url",
"browser"
],
"filePattern": "access*.log",
"timeFormat": "%d/%b/%Y:%H:%M:%S",
"filterRegex": []
},
"outputType": "LogService",
"configName": "logtail-config-sample"
}
```

Response example:

```
{
"date": "Mon, 09 Nov 2015 09:14:32 GMT",
"connection": "close",
"x-log-requestid": "564063F899248CAA2300B778",
"content-length": "0",
"server": "nginx/1.6.1"
}
```

RAM subaccount access

Overview

Use RAM to access the primary account’ s Log Service resources from a subaccount

The project, LogStore, config, and machinegroup you create are your own resources. By default, you have the full operation permissions on your resources, and can use all APIs described in this document to perform operations on your resources.

However, in scenarios where a primary account has a subaccount, you cannot use an unauthorized subaccount to perform operations on resources of the primary account. You need to grant the permission to the subaccount to perform operations on resources of the primary account through RAM authorization.

> **NOTE:** Before learning how to use RAM to authorize a subaccount and access resources of a primary account, ensure that you have carefully read RAM product documentation and API documentation.

Three authorization policies for Log Service are available on RAM console:

AliyunLogFullAccess

This policy is used to grant a subaccount the full permission to access Log Service resources of a primary account. The authorization policy is described as follows:

```
 {
"Version": "1",
"Statement": [
{
"Action": "log:*",
"Resource": "*",
"Effect": "Allow"
}
]
}
```

AliyunLogReadOnlyAccess

This policy is used to grant a subaccount the read-only permission for Log Service resources of a primary account. The authorization policy is described as follows:

```
 {
"Version": "1",
"Statement": [
{
"Action": [
"log:Get*",
"log:List*"
],
"Resource": "*",
"Effect": "Allow"
}
]
}
```

Query data of a specified LogStore on the console

This policy is used to grant a subaccount that logs into the console the read-only permission for the specified LogStore resources of a primary account (view and extract logs, and view the LogStore list). The authorization policy is described as follows:

```
 {
"Version": "1",
"Statement": [
{
```

```
"Action": ["log:ListProject", "log:ListLogStores"],
"Resource": ["acs:log:*:*:project/<Name of the specified project>/*"],
"Effect": "Allow"
} ,
{
"Action": ["log:Get*"],
"Resource": ["acs:log:*:*:project/<Name of the specified project>/logstore/<Name of the specified
LogStore>"],
"Effect": "Allow"
}
]
}
```

If you do not need to grant an account the permission to access Log Service resources in another account, you can skip this section. Skipping this section does not affect your understanding and use of the remaining parts in the file.

More information:

- Types of Log Service resources in RAM that can be authorized
- Actions in RAM that can be performed on a Log Service resource
- Rules for authenticating a subaccount’ s access to the primary account’ s resources through Log Service APIs

Resource list

Types of Log Service resources in RAM that can be authorized

The following table lists the types of resources that can be authorized in RAM and how they are described:

Resource type	Resource description in authorization policy
Project/Logstore	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}
Project/Logstore/Shipper	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/shipper/${shipperName}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/*
Project/Config	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailconfig}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/*

Project/MachineGroup	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/*
Project/ConsumerGroup	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/*
Generic mode	acs:log:${regionName}:${projectOwnerAliUid}:*
	acs:log:*:${projectOwnerAliUid}:*

NOTE: There is a hierarchical relationship among Log Service resources. Project is a top-level resource. LogStore, config, and machinegroup are at the same level and sub-resources of the project. LogShipper and consumergroup are sub-resources of the LogStore.

Wherein:

- ${regionName} indicates the name of a region.
- ${projectOwnerAliUid} indicates the user’ s Alibaba Cloud account ID.
- ${projectName} indicates the name of a Log Service project.
- ${logstoreName} indicates the name of a LogStore.
- ${logtailconfig} indicates the name of a config.
- ${machineGroupName} indicates the name of a machinegroup.
- ${shipperName} indicates the name of a LogShipper rule.
- ${consumerGroupName} indicates the name of a consumer group.

Action list

Actions in RAM that can be performed on a Log Service resource

In RAM, you can perform the following actions on a Log Service resource. Each action corresponds to one or two APIs.

- log:GetLogStore
- log:ListLogStores
- log:CreateLogStore

- log:DeleteLogStore
- log:UpdateLogStore
- log:GetCursorOrData (GetCursor，PullLogs)
- log:ListShards
- log:PostLogStoreLogs
- log:CreateConfig
- log:UpdateConfig
- log:DeleteConfig
- log:GetConfig
- log:ListConfig
- log:CreateMachineGroup
- log:UpdateMachineGroup
- log:DeleteMachineGroup
- log:GetMachineGroup
- log:ListMachineGroup
- log:ListMachines
- log:ApplyConfigToGroup
- log:RemoveConfigFromGroup
- log:GetAppliedMachineGroups
- log:GetAppliedConfigs
- log:GetShipperStatus
- log:RetryShipperTask
- log:CreateConsumerGroup
- log:UpdateConsumerGroup
- log:DeleteConsumerGroup
- log:ListConsumerGroup
- log:ConsumerGroupUpdateCheckPoint
- log:ConsumerGroupHeartBeat
- log:GetConsumerGroupCheckPoint

Authentication rules

Rules for authenticating a subaccount’ s access to the primary account’ s resources through Log Service APIs

When a subaccount accesses the primary account’ s resources through Log Service Open API , the Log Service background performs RAM permission inspection to ensure that the resource owner has granted relevant permissions to the caller.

Different Log Service APIs will determine the access to which resources needs to be checked according to the involved resources and the meanings of the API. Specifically, the authentication rules for each API are described in the table below:

logstore

Action	Resource
log:GetLogStore	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/**${logstoreName}**
log:ListLogStores	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/*
log:CreateLogStore	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/*
log:DeleteLogStore	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/**${logstoreName}**
log:UpdateLogStore	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/**${logstoreName}**

loghub

The rule is applicable to APIs for data writing and consumption. The API GetCursor for getting data cursor and API GetLogs for getting data share the same action (log:GetCursorOrData).

Action	Resource
log:GetCursorOrData	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/**${logstoreName}**
log:ListShards	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/**${logstoreName}**
log:PostLogStoreLogs	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logstore/**${logstoreName}**

config

Action	Resource
log:CreateConfig	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/*
log:UpdateConfig	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/**${logtailConfigName}**
log:DeleteConfig	acs:log:**${regionName}**:**${projectOwnerAliUid}**:

	project/**${projectName}**/logtailconfig/**${logtailConfigName}**
log:GetConfig	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/**${logtailConfigName}**
log:ListConfig	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/*

machinegroup

Actions	Resources
log:CreateMachineGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/*
log:UpdateMachineGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**
log:DeleteMachineGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**
log:GetMachineGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**
log:ListMachineGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/*
log:ListMachines	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**

APIs for config and machinegroup interaction

Actions	Resources
log:ApplyConfigToGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/**${logtailConfigName}** acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**
log:RemoveConfigFromGroup	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/**${logtailConfigName}** acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**
log:GetAppliedMachineGroups	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/logtailconfig/**${logtailConfigName}**

log:GetAppliedConfigs	acs:log:**${regionName}**:**${projectOwnerAliUid}**:project/**${projectName}**/machinegroup/**${machineGroupName}**

STS access mode

Use STS for Cross-account Log Resource Access

The project, LogStore, config, and machinegroup you create are your own resources. By default, you have the full operation permissions on your resources, and can use all APIs described in this document to perform operations on your resources.

To authorize another account to access your resources, you must use the STS to get a temporary AK/token for performing specific operations. Before reading the following instructions, read the STS product documentation.

Assume that User A creates a project, LogStore, and other resources in Log Service, and User B wants to call an API to access these resources, the operation steps are as follows.

Operations of User A

Create a role

User A creates a role for User B’ s cloud account through access control service console or API. The role details are described below:

```
{
"Statement": [
{
"Action": "sts:AssumeRole",
"Effect": "Allow",
"Principal": {
"RAM": [
"acs:ram::<ID of the cloud account B>:root"
]
}
}
],
"Version": "1"
}
```

Role authorization

After a role is created, User A needs to grant specific operation permissions for the role.

Permissions required for data writing only are as follows:

```
```
{
"Version": "1",
"Statement": [
{
"Action": "log:PostLogStoreLogs",
"Resource": "*",
"Effect": "Allow"
}
]
}
```
```

Permissions required for data extraction by using the collaborative consumer group are as follows:

```
```
{
"Version": "1",
"Statement": [
{
"Action": [
"log:GetCursorOrData",
"log:CreateConsumerGroup",
"log:ListConsumerGroup",
"log:ConsumerGroupUpdateCheckPoint",
"log:ConsumerGroupHeartBeat",
"log:GetConsumerGroupCheckPoint"
]
"Resource": "*",
"Effect": "Allow"
}
]
}
```
```

How to set a resource is described below:

- The above two types of resources mean that the specified user’ s all projects and LogStores are authorized.
- To authorize a specific project: acs:log::*{projectOwnerAliUid}:project/*-To authorize a specific LogStore: acs:log::*{projectOwnerAliUid}:project/{projectName}/logstore/{logstoreName}/*

For complete resource description, refer to Log Service RAM resources.

Operations of User B

Create and authorize a subaccount

Log into the access control service console or use API/SDK to create a subaccount, and grant the AssumeRole permission to the subaccount.

Call an STS interface to get a temporary AK/token

STS SDK usage instructions

Call a Log Service Interface

Log Service SDk usage instructions

Sample code

The sample code is applicable to the case where User B writes data to a project of User A through STS (based on JavaSDK).

Code link

Common resources

Data model

To facilitate understanding and use of Log Service, the following describes some basic concepts.

Region

A region is a service node of Alibaba Cloud. By deploying services in different Alibaba Cloud regions, you can make your services closer to your clients for lower access latency and better user experience. Alibaba Cloud has multiple regions throughout the country.

Project

The project is a basic management unit in Log Service and is used for resource isolation and control. You can use a project to manage all logs and related log sources of one application.

Logstore

LogStore is a unit collecting, storing, and consuming log data in Log Service. Each logstore belongs to one project, and multiple logstores can be created for each project. You can create multiple logstores for one project according to actual needs. The common practice is to create an independent logstore for each type of log in one application. For example, assume that you have a game application named big-game, and there are three types of logs on the server: operation_log, application_log, and access_log. You can first create a project named big-game, and then create three logstores for the three types of logs under this project for log collection, storage and consumption, respectively.

Log

A log is a minimum data unit processed in Log Service. Log Service uses a semi-structured data mode to define a log. The specific data model is as follows:

- **Topic**: a custom field to mark a batch of logs (for example: access_logs are marked according to sites). This field is a null string by default (the null string is also a valid topic).
- **Time**: a reserved field in the log, which is used to indicate the generation time of the log (precise to the second, calculated in seconds from 1970-1-1 00:00:00 UTC) and is generally generated directly based on the time in the log.
- **Content**: used to record the specific content of the log. Content is composed of one or more content items, and each content item is composed of a Key-Value pair.
- **Source**: source of the log, for example, the IP address of the device generating the log. This field is null by default.

Furthermore, Log Service has different requirements on values of different fields, as described in the following table:

Data Field	Requirement
time	Integer, standard time format of Unix. The minimum unit is second.
topic	Any UTF-8 encoded string of no more than 128 bytes.
source	Any UTF-8 encoded string of no more than 128 bytes.
content	One or more Key-Value pairs. Key is a UTF-8 encoded string of no more than 128 bytes, which contains only letters, underlines, and numbers and cannot begin with a number. Value is any UTF-8 encoded string of no more than 1024*1024 bytes.

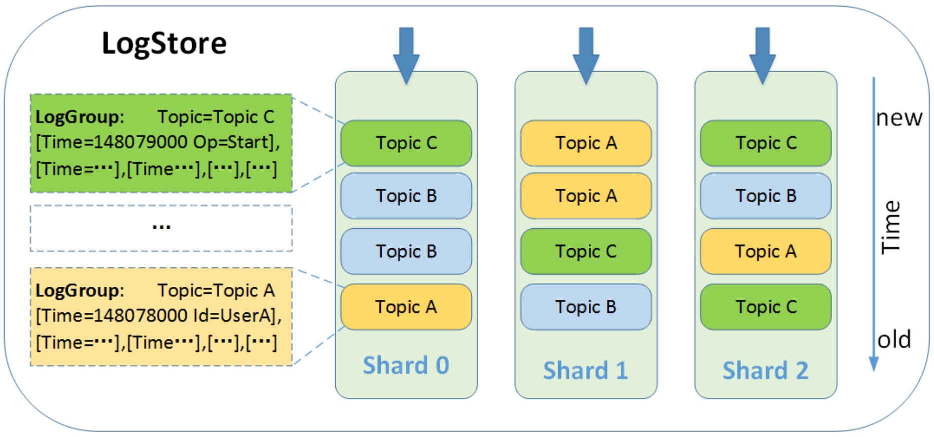
The following keywords cannot be used in the key in content described in the preceding table: __time__, __source__, __topic__, __partition_time__, _extract_others_, and __extract_others__.

Log topic

Logs in the same LogStore can be grouped by log topics. You can specify the topic when writing a log. For example, a platform user can use the user ID as the log topic and write it into the log. If there is no need to group the logs in one logstore, the same log topic can be used for all logs.

NOTE: A null string is a valid log topic. The default log topic is a null string.

The following diagram describes the relation among Logstore, log topic, and log:


LogStore
LogGroup: Topic=Topic C [Time=148079000 Op=Start], [Time=···],[Time···],[···],[···]
...
LogGroup: Topic=Topic A [Time=148078000 Id=UserA], [Time=···],[Time···],[···],[···]
Topic C
Topic B
Topic B
Topic A
Shard 0
Topic A
Topic A
Topic C
Topic B
Shard 1
Topic C
Topic B
Topic A
Topic C
Shard 2
new
Time
old

Various log formats are used in actual application scenarios. For ease of understanding, the following describes how to map an original Nginx access_log to the log data model in Log Service. Assume that the IP address of your Nginx server is 10.249.201.117, and the following is the original log:

```
10.1.168.193 - - [01/Mar/2012:16:12:07 +0800] "GET /Send?AccessKeyId=8225105404 HTTP/1.1" 200 5 "-"
"Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2"
```

Map the original log to the log data model in Log Service as follows:

Data Field	Content	Description
topic	“”	Use the default value (null string).
time	1330589527	Precise generation time of the log (precise to the second), which is transformed from the time stamp in the original log.
source	“10.249.201.117”	Use the IP address of the server as the log source
content	Key-Value 对	Content of the log

You can decide how to extract the original content of the log and combine it into Key-Value pairs. For example, see the following table:

key	value
ip	“10.1.168.193”
method	“GET”
status	“200”
length	“5”
ref_url	“- ”
browser	“Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firef

Logs

A collection of multiple logs.

LogGroup

A group of logs.

LogGroupList

A group of LogGroups used for return of the results.

Encoding method

Currently, the system supports the following content encoding method (scalable in the future). The Restful API layer is indicated by Content-Type.

	Meaning	Content-Type
ProtoBuf	The data model is encoded by ProtoBuf.	application/x-protobuf

The following PB defines the object of the data model:

```
message Log
{
required uint32 Time = 1;// UNIX Time Format
message Content
{
required string Key = 1;
required string Value = 2;
}
repeated Content Contents= 2;
}
message LogGroup
{
repeated Log Logs= 1;
optional string Reserved = 2; // reserved fields
optional string Topic = 3;
optional string Source = 4;
}
message LogGroupList
{
repeated LogGroup logGroupList = 1;
}
```

NOTE: Because PB does not require uniqueness of the Key-Value pair, you need to avoid such case. Otherwise, the behavior is undefined.

Data coding mode

Protocol Buffer is a structured data interchange format developed by Google. It is widely used in many internal and external services of Google. Log Service uses Protocol Buffer as the standard format in writing logs. You need to serialize the original log data into Protocol Buffer data streams before writing [Logs] (~~28961~~) into the server by using API.

```

message Log
{
required uint32 time = 1; // UNIX Time Format
message Content
{
required string key = 1;
required string value = 2;
}
repeated Content contents= 2;
}

message LogGroup
{
repeated Log logs= 1;
optional string reserved =2; // Internal field, which does not need to be specified
optional string topic = 3;
optional string source = 4;
}

message LogGroupList
{
repeated LogGroup logGroupList = 1;
}
```

Note:

- Because PB does not require uniqueness of the Key-Value pair, you need to avoid such case. Otherwise, the behavior is undefined.
- For more details about Protocol Buffer, go to Github Home.
- For details about the APIs for log write function in Log Service, refer to [PostLogStoreLogs] (~~29026~~).

LogStore

The data storage unit is called LogStore. By default, you can create 10 logstores for each project. The LogStore name is unique in the project. LogStore is an ingress for all log data. You can perform read and write operations on the logstore.

LogStore naming rules:

- A LogStore name can only contain lower-case letters, numbers, hyphens (-), and underlines (_).
- A LogStore name must start and end with a lower-case letter or number.
- The length must be 2–63 bytes.

Example of the complete resource:

```
{
  “logstoreName" : "access_log",
"ttl": 1,
"shardCount": 2,
"createTime":1439538649,
"lastModifyTime":1439538649
}
```

Parameter definitions:

Parameter Name	Type	Required or Not	Description
logstoreName	string	Yes	The logstore name, which is unique in the project.
ttl	integer	Yes	The Time-to-Live (TTL) of log data. The unit is days and the minimum value is 1 day.
shardCount	integer	Yes	Log data Service Unit
createTime (OutputOnly)	integer	No	The creation time of this resource on the server (output only).

lastModifyTime (OutputOnly)	integer	No	The update time of this resource on the server (output only).

Shard

A shard is a basic read and write unit in each logstore. You can specify the number of shards in each logstore. Each shard has certain service capabilities:

- Write:5MB/s
- Read:10MB/s

When writing data into a shard, you must specify the shard. When reading data, you can use the Load-Balance mode. Load-Balance automatically performs balance based on the background system load, to ensure high write availability.

Example of the complete resource:

Parameter Name	Type	Required or Not	Description
shardID	int	Yes	The unique ID of shard in the logstore, which is automatically generated by the system and whose type is integer.

logtail configuration

Config is the configuration of logtail. By default, you can create up to 100 configs for each project. A config name must be unique in a project.

You can use config to specify the location, method, and parameters for collecting the logs.

Config naming rules:

- A config name can only contain lower-case letters, numbers, hyphens (-), and underlines (_).
- A config name must start and end with a lower-case letter or number.
- The length must be 2–128 bytes.

Example of the complete resource:

```
{
```

```
"configName": "testcategory1",
 “inputType” : ” file” ,
"inputDetail": {
 “logType” :  “common_reg_log” ,
 “logPath” :  “/var/log/httpd/” ,
 “filePattern” :  “access.log” ,
 “localStorage” : true,
 “timeFormat” :  “%Y/%m/%d %H:%M:%S” ,
 “logBeginRegex” :  “.*” ,
 “regex” :  “(\w+)(\s+)” ,
 “key”  :[ “key1” ,  “key2” ],
 “filterKey” :[ “key1” ],
 “filterRegex” :[ “regex1” ],
 “topicFormat” :  “none”
},
 “outputType” : ” sls” ,
 “outputDetail” :
{
 “logstoreName” : ” perfcounter”
},
 “createTime": 1400123456,
 “lastModifyTime": 1400123456
}
```

Attribute Name	Type	Required or Not	Description
configName	string	Yes	The log config name, which is unique in the project.
inputType	string	Yes	The input type, which is “file” by default.
inputDetail	json	Yes	See description in the following table.
outputType	string	Yes	The output type. Currently, only LogService is supported.
outputDetail	string	Yes	See description in the following table.
createTime(output-only)	integer	No	The creation time of config.
lastModifyTime (output-only)	integer	No	The update time of this resource on the server.

Content of inputDetail:

Attribute Name	Type	Required or Not	Description
logType	string	Yes	The log type. Currently, only common_reg_log is supported.
logPath	string	Yes	The parent directory where the log is located, for example, /var/logs/.
filePattern	string	Yes	The pattern of the log file, for example, access*.log.
localStorage	boolean	Yes	Whether to activate local cache. Logs of 1 GB can be stored locally when the links across the servers are disconnected.
timeFormat	string	Yes	The format of log time, for example, %Y/%m/%d %H:%M:%S.
logBeginRegex	string	Yes	The characteristics (regular expression) of the first log line, used in the case of a log composed of multiple lines.
regex	string	Yes	The regular expression used for extracting a log pair.
key	array	Yes	The key generated after the log is extracted.
filterKey	array	Yes	The key used for filtering the log. The log meets requirements only when the key value matches the regular expression specified in the corresponding filterRegex column.
filterRegex	array	Yes	The regular expression corresponding to each filterKey. The length of filterRegex

			must be same as that of filterKey.
topicFormat	string	No	This is used to take one part of a log file path as a topic, for example, /var/log/(.*).log. The default value is none, indicating that the topic is null.
preserve	boolean	No	“true” indicates that the monitored directories will never be timed out and “false” indicates that the monitored directories have been timed out by 30 minutes. The default value is true.
preserveDepth	integer	No	If preserve is set to false, the depth of the directories with no monitoring timeout is specified. The maximum depth is 3.

Content of outputDetail:

Attribute Name	Type	Required or Not	Description
logstoreName	string	Yes	Name of the LogStore.

logtail machine group

Machine: After logtail is installed and the machine is started normally, the machine is automatically associated with the current user based on the user information in the logtail configurations. Currently, the machine can be identified in three ways:

- IP: The IP address corresponding to the hostname. This is the easiest method to understand, but the IP address may be duplicated in the VPC and other environments.
- UUID (machine-uniqueid)： The UUID in DMI devices. For details, refer to RFC4122.
- Userdefined-id: You can define machine identification in the logtail directory.

Attributes of each machine are as follows:

```
{
"ip" : "testip1",
"machine-uniqueid" : "testuuid1",
"userdefined-id" : "testuserdefinedid1",
"lastHeartbeatTime” :1397781420
}
```

Parameter Name	Type	Description
ip	string	The IP address corresponding to the machine hostname.
uuid	string	The unique primary key for machine identification, which is uploaded by logtail.
userdefined-id	string	The custom machine ID uploaded by logtail.
lastHeartbeatTime(output-only)	integer	The last heartbeat time of the machine (the seconds starting from the epoch time).

machinegroup

Machinegroup is the machine group that belongs to the current user in the project. The machinegroup can be identified in two ways (IP address and userdefined). The IP address is more easily identified while the “userdefined” method can solve the problem of identical IP address on the VPC. You can select either of the two machine identification methods.

Machinegroup naming rules:

- A config name can only contain lower-case letters, numbers, hyphens (-), and underlines (_).
- A config name must start and end with a lower-case letter or number.
- The length must be 2–128 bytes.

Example of the complete resource

```
{
"groupName" : "testgroup",
"groupType" : "",
"groupAttribute" : {
"externalName" : "testgroup",
 “groupTopic” : “testgrouptopic”
},
 “machineIdentifyType” : "ip",
"machineList" : [
"ip1",
"ip2"
...
],
  “createTime”  : 1431705075,
"lastModifyTime" : 1431705075
}
```

Attribute Name	Type	Required or Not	Description
groupName	string	Yes	The machinegroup name, which is unique in a project.
groupType	string	No	The machine group type, which is empty by default
machineIdentifyType	string	Yes	The machine identifier type, including IP and userdefined
groupAttribute	object	Yes	The machine group attribute, which is empty by default
machineList	array	Yes	The specific machine ID, which can be an IP address or a user-defined ID
createTime(output-only)	int	No	The creation time of this resource.
lastModifyTime(output-only)	int	No	The update time of this resource on the server.

Group attributes are described as follows:

Name	Type	Required or Not	Description
groupTopic	string	No	The topic of a machine group, which is empty by default
externalName	string	No	The external management ID, which is empty by default