Log Service

Developer Guide

MORE THAN JUST CLOUD | Alibaba Cloud

Developer Guide

API Reference

Overview

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

Besides the Log Service console, you can 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	Basic concepts of logs and log groups
Config	List, Create, Delete, Get, and Update
	GetAppliedMachineGroups (query the applied machine groups)
MachineGroup	List, Create, Delete, Get, and Update
	Apply/Remove (apply/remove a configuration)
	GetAppliedConfigs (query the list of applied configurations)
Logstore	List, Create, Delete, Get, and Update
	GetLogs (query logs) and GetHistograms (query log distribution)
Shard	List, Split, and Merge
	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:

- Collect logs based on configurations and machine groups.
- Create a Logstore. Then, write and read logs to/from the Logstore.
- Set access control rules for different users.

Note:

- Currently, APIs provide the RESTful style.
- To use APIs, you must know the API access address.
- Security verification is required for all API requests. For more information about the signature and process of API requests, see Request signature.
- Log Service supports Resource Access Management (RAM) and Security Token Service (STS). Similar to common cloud accounts, RAM sub-accounts can use APIs by using their AccessKey signature. To use the STS temporary identity, you must use the temporary AccessKey and enter a special HTTP header. For more information, see Public request header. This HTTP header must participate in the signature. For more information, see Request signature.

Service endpoint

Internet service endpoint

The Log Service endpoint is a URL used to access a project and logs within the project, and is associated with the Alibaba Cloud region where the project resides and the project name. Currently, Log Service has been activated in multiple Alibaba Cloud regions. The Internet service endpoints for each region are as follows.

Region	Service endpoint
China East 1 (Hangzhou)	cn-hangzhou.log.aliyuncs.com
China East 2 (Shanghai)	cn-shanghai.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 North 5 (Huhehaote)	cn-huhehaote.log.aliyuncs.com
China South 1 (Shenzhen)	cn-shenzhen.log.aliyuncs.com
Hong Kong (China)	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
Asia Pacific SE 3 (Kuala Lumpur)	ap-southeast-3.log.aliyuncs.com
Asia Pacific SE 5 (Jakarta)	ap-southeast-5.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
US East 1 (Virginia)	us-east-1.log.aliyuncs.com
Asia Pacific SOU 1 (Mumbai)	ap-south-1.log.aliyuncs.com

When accessing a specific project, you must give a final access address composed of the project name and the region where the project resides. 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, then the access address is as follows:

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

Note: You must specify a region when creating a Log Service project. After the project is created, you cannot modify the region or migrate the project across regions, and you must select a root service endpoint address that matches the region to compose the access address for this project. The service endpoint is used for API requests.

Classic network/VPC service endpoint

To use Log Service APIs on an Alibaba Cloud Elastic Compute Service (ECS) instance (including the Virtual Private Cloud (VPC) environment), you can also use the intranet service endpoints. Using intranet service endpoints to access Log Service does not consume ECS Internet traffic and saves the valuable ECS public network bandwidth). The intranet root service endpoints for each region are as follows.

Region	Root service endpoint

China East 1 (Hangzhou)	cn-hangzhou-intranet.log.aliyuncs.com
China East 2 (Shanghai)	cn-shanghai-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 North 3 (Zhangjiakou)	cn-zhangjiakou-intranet.log.aliyuncs.com
China North 5 (Huhehaote)	cn-huhehaote-intranet.log.aliyuncs.com
Hong Kong (China)	cn-hongkong-intranet.log.aliyuncs.com
US West 1 (Silicon Valley)	us-west-1-intranet.log.aliyuncs.com
Asia Pacific NE 1 (Tokyo)	ap-northeast-1-intranet.log.aliyuncs.com
Asia Pacific SE 1 (Singapore)	ap-southeast-1-intranet.log.aliyuncs.com
Asia Pacific SE 2 (Sydney)	ap-southeast-2-intranet.log.aliyuncs.com
Asia Pacific SE 3 (Kuala Lumpur)	ap-southeast-3-intranet.log.aliyuncs.com
Asia Pacific SE 5 (Jakarta)	ap-southeast-5-intranet.log.aliyuncs.com
Middle East 1 (Dubai)	me-east-1-intranet.log.aliyuncs.com
EU Central 1 (Frankfurt)	eu-central-1-intranet.log.aliyuncs.com
US East 1 (Virginia)	us-east-1-intranet.log.aliyuncs.com
Asia Pacific SOU 1 (Mumbai)	ap-south-1-intranet.log.aliyuncs.com

For example, if the project name is big-game and it is in the China East 1 (Hangzhou) region, then the intranet access address is as follows:

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

Note: Currently, Log Service APIs in the preceding service endpoints only support the HTTP or HTTPS protocol.

AccessKey

Alibaba Cloud AccessKey is a “secure password” designed for you to access your cloud resources by using APIs (not the console). You can use the AccessKey to sign API request content to pass the security authentication in Log Service.

This AccessKey is generated and used by pairing an AccessKey ID and an AccessKey Secret. Each Alibaba Cloud user can create multiple AccessKeys. You can also activate, deactivate, or delete the generated AccessKey as per your needs.

You can create and manage all the AccessKeys on the **Access Key Management** page in the Alibaba Cloud console. Keep your AccessKey properly because it is key to the API request security authentication of Alibaba Cloud. We recommend that you delete the AccessKey in time and generate a new one if the AccessKey may have been leaked.

Public request header

Log Service APIs are RESTful APIs based on the HTTP protocol, which support 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). See the following detailed definitions.

Header name	Type	Description
Accept	string	The type that the client expects Log Service to return. Currently, application/json and application/x-protobuf are supported. This field is optional and valid only for GET requests. The specific value is subject to the definition of each API.
Accept-Encoding	string	The compression algorithm that the client expects Log Service to return. Currently, lz4, deflate, and null (not compressed) are supported. This field is optional and valid only for GET requests. The specific value is subject to the definition of each API.
Authorization	string	The signature content. For more information, see Request signature.
Content-Length	numeric value	The length of the HTTP request body defined in RFC 2616. If the request has no body, this request header is not required.
Content-MD5	string	The string generated after the request body undergoes MD5 computing, and the

		computing result is in uppercase. If the request has no body, this request header is not required.
Content-Type	string	The type of the HTTP request body defined in RFC 2616. Currently, Log Service API requests only support application/x-protobuf. If the request has no body, this request header is not required. The specific value is subject to the definition of each API.
Date	string	The time when the request is sent. Currently, parameters only support 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	The complete host name of the HTTP request, which does not include protocol headers such as http://. For example, big-game.cn-hangzhou.sls.aliyuncs.com.
x-log-apiversion	string	The API version. The current version is 0.6.0.
x-log-bodyrawsize	numeric value	The initial size of the request body. If the request has no body, the value is 0. If the body is compressed data, the value is the size of the raw data before the compression. The value range for this field is 0–3x1024x1024. This field is optional and only required when the body is compressed.
x-log-compresstype	string	The compression type of the API request body. Currently, lz4 and deflate are supported (RFC 1951 uses the zlib format. For more information, see RFC 1950). If the body is not compressed, this request header is not required.
x-log-date	string	The time when the request is

		sent. The format is the same as that of the Date header. This request header is optional. If a request contains this public request header, this value will replace the value of the standard header Date for request authentication in Log Service. This field does not participate in the signature. Whether or not the x-log-date header exists, you must provide the HTTP standard header Date.
x-log-signaturemethod	string	The signature computing method. Currently, only hmac-sha1 is supported.
x-acs-security-token	string	Use the Security Token Service (STS) temporary identity to send data. This request header is required only when the STS temporary identity is used.

Note:

- The maximum difference between the time expressed in the Date header of a request 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 computed based on the value of the x-log-date header.
- If the compression algorithm is specified in x-log-compresstype of the request, the raw data must be compressed and then put into the HTTP body. The Content-Length and Content-MD5 headers are computed based on the compressed body.
- The Date header cannot be specified when HTTP requests are sent from some platforms (the request sent time is automatically specified by the platform library). Therefore, the correct Date value cannot be used to compute the request signature. In this situation, specify the x-log-date header and use this request header value to compute the request signature. After receiving an API request, Log Service first determines whether or not the request contains an x-log-date header. If yes, Log Service uses the header value for signature authentication. Otherwise, Log Service uses the HTTP standard header Date for signature authentication.

Public response header

Log Service APIs are RESTful APIs based on the HTTP protocol. All the Log Service API responses provide a set of public response headers. See the following detailed definitions.

Header name	Type	Description
Content-Length	numeric value	The length of the HTTP response content defined in RFC 2616.
Content-MD5	string	The MD5 value of the HTTP response content defined in RFC 2616, which is an uppercase string generated after the body undergoes MD5 computing.
Content-Type	string	The type of the HTTP response content defined in RFC 2616. Currently, Log Service supports two response types: application/json and application/x-protobuf.
Date	string	The time when the request is returned. Currently, parameters only support 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	The unique ID generated in Log Service that marks this request. This response header is not related to specific applications, but is mainly used to track and investigate problems. To troubleshoot the API request that fails, provide this ID to the Log Service team.

Request signature

To guarantee the security of your logs, all the HTTP requests of Log Service APIs must pass the security authentication. Currently, this security authentication is based on the Alibaba Cloud AccessKey and is completed by using the symmetric encryption algorithm.

The process is as follows:

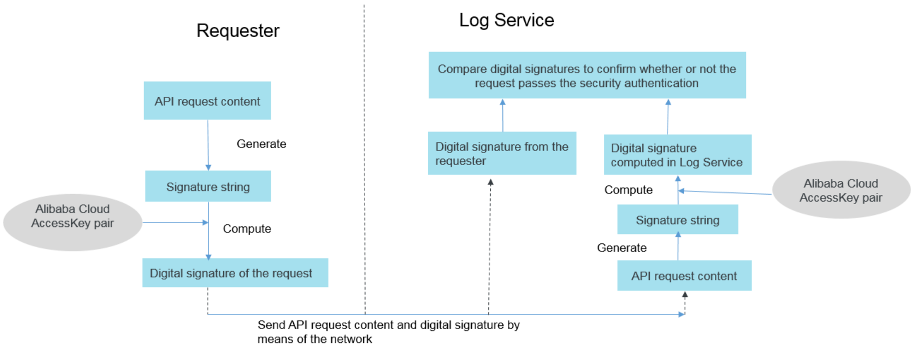
1. The requester generates a signature string based on the API request content (including the HTTP header and body).
2. The requester uses Alibaba Cloud AccessKey pair (AccessKey ID and AccessKey Secret) to sign the signature string generated in the first step, forming a digital signature for this API request.
3. The requester sends both the API request content and digital signature to Log Service.

   After receiving the request, Log Service repeats steps 1 and 2 and computes the expected digital signature for this request.

   > **Note:** Log Service retrieves the AccessKey pair used by this request from the backend.

   Log Service compares the expected digital signature and the digital signature sent from the requester. If they are the same, the request passes the security authentication. Otherwise, the request is rejected directly.

The entire process can be intuitively described by the following diagram.


Requester
Log Service
Compare digital signatures to confirm whether or not the request passes the security authentication
API request content
Generate
Digital signature from the requester
Digital signature computed in Log Service
Signature string
Compute
Alibaba Cloud AccessKey pair
Alibaba Cloud AccessKey pair
Compute
Signature string
Generate
Digital signature of the request
API request content
Send API request content and digital signature by means of the network

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

- Confirm which user sends the API request. This is because a user must specify the AccessKey pair used to generate the digital signature before sending the request, and Log Service can then confirm the user identity by using this AccessKey pair and manage the access permission.
- Confirm whether or not the user request is tampered during the network transmission. This is because Log Service recomputes the digital signature for the received request content. If the request content is tampered during the network transmission, the digital signature cannot match.

Sign an API request

To pass the API request security authentication, you must sign this API request in the client (namely, generate a correct digital signature) and use the HTTP header Authorization to transmit the digital signature of this request by means of the network. The specific format of the Authorization header is as follows:

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

As shown in the preceding format, the Authorization header value contains the AccessKey ID of the AccessKey pair and the corresponding AccessKey Secret is used to construct the signature value. Follow these steps to construct the signature value:

Step 1: Prepare a suitable Alibaba Cloud AccessKey pair

To generate a digital signature for an API request, you must use an AccessKey pair (AccessKey ID and AccessKey Secret). You can use an existing AccessKey pair or create a new one. Make sure the used AccessKey pair is enabled.

Step 2: Generate the signature string of the request

The signature string of a Log Service API is generated by using the method, header, and body of the HTTP request. See the detailed generation method as follows:

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

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

Name	Definition	Example
VERB	The method name of the HTTP request.	PUT, GET, and POST
CONTENT-MD5	The MD5 value of the HTTP request body, which must be an uppercase string.	875264590688CA6171F6228AF5BBB3D2
CONTENT-TYPE	The type of the HTTP request body.	application/x-protobuf
DATE	The standard timestamp header of the HTTP request, which follows the RFC 1123 format and uses the GMT standard time.	Mon, 3 Jan 2010 08:33:47 GMT

CanonicalizedLOGHeaders	The string constructed by custom headers prefixed by x-log and x-acs in the HTTP request (for the specific construction method, see the following description).	x-log-apiversion:0.6.0\nx-log-bodyrawsize:50\nx-log-signaturemethod:hmac-sha1
CanonicalizedResource	The string constructed by the HTTP request resources (for the specific construction method, see the following description).	/logstores/app_log

For HTTP requests without the body, the CONTENT-MD5 and CONTENT-TYPE fields are empty strings. The generation method of the signature string is as follows:

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

> **Note:** As described in Public request headers, the custom request header x-log-date is introduced to Log Service APIs. If you specify this header in your request, the header value will replace the value of the HTTP standard request header Date to compute the request signature.

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. Sort all Log Service custom request headers obtained in the previous step lexicographically in ascending order.
3. Delete any space at either side of a separator between request header and content.
4. Separate all headers and contents with 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 Service resources to be accessed. For example, /logstores/logstorename. The field is left blank if the logstorename does not exist.
3. If the request contains a query string (QUERY_STRING), add question mark (?) and the query string at the end of the CanonicalizedResource string.

The QUERY_STRING is a string generated after the request parameters in the URL are sorted lexicographically. Use an equal sign (=) between the parameter name and the parameter value to form a string. Sort the parameter name - value pairs lexicographically in ascending order. Then, use &

to connect the pairs to form a string. The formula is as follows:

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

Step 3: Generate the digital signature of the request

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

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

> **Note:** Use the HMAC-SHA1 method defined in RFC 2104 as the signature method. The AccessKey Secret used in the preceding formula must correspond to the AccessKey ID used in the final Authorization header. Otherwise, the request cannot pass the authentication in Log Service.

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

Examples of the request signature process

To better understand the complete request signature process, use two examples to demonstrate the process. First, assume that the AccessKey pair used for Log Service API signature is as follows:

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

Example 1:

To send the following GET request to list all the Logstores in the ali-test-project 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 signature string generated by the preceding Log Service API request is as follows:

```
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
```

As a GET request, this request has no HTTP body. Therefore, the CONTENT-TYPE and CONTENT-MD5 fields in the generated signature string are empty strings. To use the previously specified AccessKey Secret to compute the request signature, the obtained signature is as follows:

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

Finally, send the following digitally signed HTTP request content:

```
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 must write the following logs to the Logstore test-logstore in the project ali-test-project.

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

Therefore, construct the following HTTP request according to the Log Service API definition:

```
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 are serialized to byte streams in the ProtoBuffer format>
```

In this HTTP request, the written log content is first serialized to the ProtoBuffer format (for more information, see ProtoBuffer format) 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 preceding signature string construction method, the signature string corresponding to this request is as follows:

```
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, use the AccessKey Secret in the preceding example to compute the request signature, the obtained signature is as follows:

```
XWLGYHGg2F2hcfxWxMLiNkGki6g=
```

Finally, send the following digitally signed HTTP request content:

```
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 are serialized to byte streams in the ProtoBuffer format>
```

Common error codes

When an API request error occurs, Log Service 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 in the following formats:

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

Among all the error messages that may be returned by Log Service, some are applicable to most of the APIs, while the others are unique to some APIs. See the following common error codes in the response of multiple APIs. For the error codes unique to some APIs, see the descriptions in the corresponding API reference.

HTTP status code	Error code	Error message	Description
411	MissingContentLength	Content-Length does not exist in http header when it	The required Content-Length request header is

		is necessary.	not 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 is not 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 compression type specified in x-log-compresstype is not supported.
400	MissingHost	Host does not exist in http header.	The HTTP standard request header Host is not provided.
400	MissingDate	Date does not exist in http heade.	The HTTP standard request header Date is not provided.
400	InvalidDateFormat	Date {date} must follow RFC822.	The Date request header value does not conform to the RFC822 standard.
400	MissingAPIVersion	x-log-apiversion does not exist in http header.	The HTTP request header x-log-apiversion is not 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 AccessKey ID is provided in the Authorization header.
401	Unauthorized	The AccessKeyId is unauthorized.	The provided AccessKey ID value is unauthorized.

400	MissingSignatureMethod	x-log-signaturemethod does not exist in http header.	The HTTP request header x-log-signaturemethod is not 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 sent time is more than 15 minutes before or after the current server time.
404	ProjectNotExist	Project {name} does not exist.	The Log Service project does not exist.
401	SignatureNotMatch	Signature {signature} is not matched.	The digital signature of the request does not match with that computed in Log Service.
403	WriteQuotaExceed	Write quota is exceeded.	The log write quota is exceeded.
403	ReadQuotaExceed	Read quota is exceeded.	The log read quota is exceeded.
500	InternalServerError	Internal server error message.	An internal server error.
503	ServerBusy	The server is busy, please try again later.	The server is busy. Try again later.

Note: The {...} in the error message indicates the specific error information. For example, {name} in the ProjectNotExist error message is replaced by a specific project name.

Logstore related APIs

CreateLogstore

Create a Logstore in a 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	Description
logstoreName	string	Yes	The Logstore name, which must be unique in the same project.
ttl	integer	Yes	The data retention time (in days).
shardCount	integer	Yes	The number of shards in this Logstore.

Request header

The CreateLogstore API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The CreateLogstore API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the CreateLogstore API may return the following special error codes.

HTTP status code	Error code	Error message
400	LogstoreAlreadyExist	logstore {logstoreName} already exists
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 the 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

Delete a Logstore, including all the shards and indexes in the Logstore.

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	Description
logstoreName	string	Yes	The Logstore name, which must be unique in the same project.

Request header

The DeleteLogstore API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The DeleteLogstore API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the DeleteLogstore API may return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does 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

Update the 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	Description
logstoreName	string	Yes	The Logstore name, which must be unique in the same project.
ttl	integer	Yes	The lifecycle (in days) of log data. The value range is 1–365. Open a ticket for additional requirements.
shardCount	integer	Yes	The number of

			shards, ranging from 1 to 10.

Request header

The UpdateLogstore API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The UpdateLogstore API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the UpdateLogstore API may return the following special error codes.

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

Detailed description

Currently, the number of shards can only be increased rather than decreased.

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

View 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	Description
logstoreName	string	Yes	The Logstore name, which must be unique in the same project.

Request header

The GetLogstore API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetLogstore API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

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

Error code

Besides the common error codes of Log Service APIs, the GetLogstore API may return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist

404	LogstoreNotExist	logstore {logstoreName} does 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

List the names of all the Logstores in 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	Description
offset(optional)	integer	No	The starting position of a returned record. The default value is 1.
size(optional)	integer	No	The maximum number of entries returned each page. The default value is 500 (the maximum value).
logstoreName	string	Yes	The Logstore name used for the request (partial matching is supported).

Request header

The ListLogstore API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The ListLogstore API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

After the ListLogstore request is successful, the response body includes the name list of all the Logstores in a specified project. The formats are as follows.

Name	Type	Description
count	integer	The number of returned Logstores.
total	integer	The total number of Logstores.
logstores	string array	The name list of returned Logstores.

Error code

Besides the common error codes of Log Service APIs, the ListLogstore API may return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does 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

List all available shards in a 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	Description
logstoreName	string	No	Logstore name

Request header

The ListShards API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

Content-Type: application/json

The ListShards API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

An array composed of shards.

```
[
{
"shardID":0
  “status” :  “readwrite",
  “inclusiveBeginKey” : "00000000000000000000000000000000",
  “exclusiveEndKey” : "8000000000000000000000000000000",
  “createTime” :1453949705
},
{
...
},
{
...
}
]
```

Detailed description

None.

Error code

Besides the common error codes of Log Service APIs, the ListShards API may return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message
400	LogStoreWithoutShard	logstore has no shard

Note: The {name} in the preceding error message is replaced by a specific 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

Split 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	Description
logstoreName	string	Yes	The Logstore name.
shardid	integer	Yes	The shard ID.
splitkey	string	Yes	The split location.

Request header

The SplitShard API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

Content-Type: application/json

The SplitShard API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

In an array composed of three shards, the first shard is the original shard before the split, and the other two are the shards generated after the split.

```
[
{
"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

None.

Error code

Besides the common error codes of Log Service APIs, the SplitShard API may return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	invalid mid hash
500	InternalServerError	Specified Server Error

		Message
400	LogStoreWithoutShard	logstore has no shard

Note: The {name} in the preceding error message is replaced by a specific 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

Merge two adjacent shards in readwrite status. Specify a shard ID in the parameter and then Log Service automatically finds the adjacent shard on the right.

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	Description
logstoreName	string	Yes	The Logstore name.
shardid	integer	Yes	The shard ID.

Request header

The MergeShards API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

Content-Type: application/json

The MergeShards API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

In an array composed of three shards, the first shard is the shard generated after the merge, and the other two are the original shards before the merge.

```
[
{
'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

None.

Error code

Besides the common error codes of Log Service APIs, the MergeShards API may return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	logstore {logstoreName} does 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

Note: The {name} in the preceding error message is replaced by a specific 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

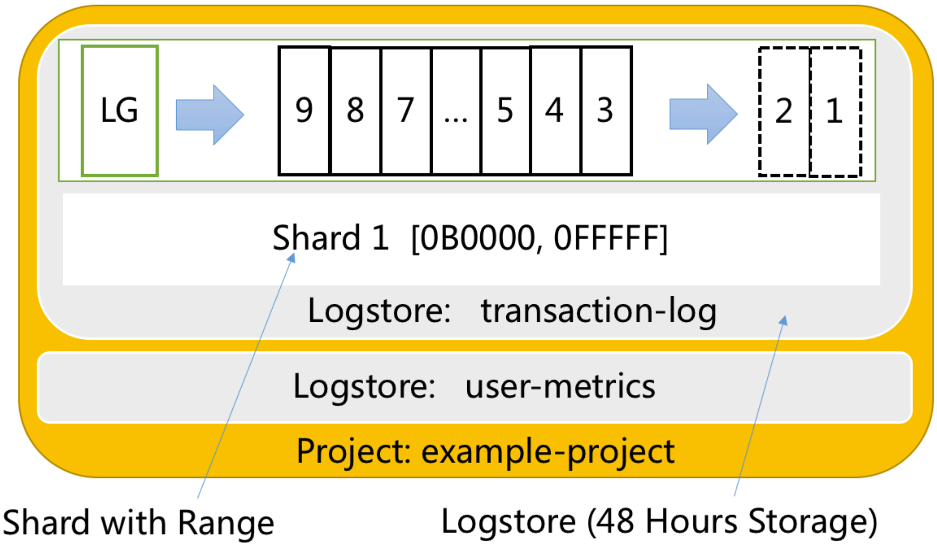
```
Header:
{
"content-length": "57",
"server": "nginx/1.6.1",
"connection": "close",
"date": "Thu, 12 Nov 2015 03:40:31 GMT",
"content-type": "application/json",
"x-log-requestid": "56440A2F99248C050600C74C"
}
Body :

[
{
'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'
}
]
```

GetCursor

The GetCursor API is used to get the cursor based on the time. The following figure shows the relationship among the project, Logstore, shard, and cursor.

- A project has multiple Logstores.
- Each Logstore has multiple shards.
- You can get the location of a specified log by using the cursor.



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	Description
shard	string	Yes	
type	string	Yes	The cursor.
from	string	Yes	The time point (in UNIX format and measured in seconds). The from_time, begin_time, or end_time.

Logstore lifecycle

The lifecycle of a Logstore is specified by the lifeCycle field in the attribute. For example, the current time is 2015-11-11 09:00:00 and lifeCycle=24. Then, the data time period that can be consumed in each shard is [2015-11-10 09:00:00,2015-11-11 09:00:00) and the time here is the server time.

By using the parameter from, you can locate the logs within the lifecycle in the shard. Assume 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

The GetCursor API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetCursor API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

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

Detailed description

N/A

Error code

Besides the common error codes of Log Service APIs, the GetCursor API may return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	Logstore {Name} does not exist
400	ParameterInvalid	Parameter From is not valid
400	ShardNotExist	Shard {ShardID} does 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

Obtain logs based on the cursor and quantity. You must specify a shard when the system obtains logs. In scenarios such as Storm, elective and collaborative consumption can be performed by using LogHubClientLib. Currently, only log group list in Protocol Buffer (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	Description
type	string	Yes	The logs.
cursor	string	Yes	A cursor used to indicate where to

			start reading data, which is equivalent to the start point.
count	integer	Yes	The number of returned log groups, ranging from 0 to 1000.

Request header

- Accept: application/x-protobuf
- Accept-Encoding: lz4, deflate, or “”

For more information about the public request headers of Log Service APIs, see Public request header .

Response header

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

For more information about the public response headers of Log Service APIs, see Public response header.

Response element

Serialized data (which may be compressed) in PB format.

Detailed description

N/A

Error code

Besides the common error codes of Log Service APIs, the PullLogs API may return the following special error codes.

HTTP status code	Error code	Error message
404	LogStoreNotExist	Logstore {Name} does not exist
400	ParameterInvalid	Parameter Cursor is not valid
400	ParameterInvalid	ParameterCount must be [0-

		1000]
400	ShardNotExist	Shard {ShardID} does not exist
400	InvalidCursor	this cursor is invalid
500	InternalServerError	Specified Server Error Message

Example

Request example

```
Read data from shard 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:
The <log group list in PB format> after the compression.
```

Page flip

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

PostLogstoreLogs

Write log data to a specified Logstore in the following modes. Currently, only log groups in Protocol Buffer (PB) format can be written.

- Load balancing mode: Automatically write logs to all writable shards in a Logstore in the load balancing mode. This mode is highly available for writing (SLA: 99.95%), applicable to scenarios in which data writing and consumption are independent of shards, for example, scenarios that do not preserve the order.
- KeyHash mode: A key is required when writing data. Log Service automatically writes data to the shard that meets the key range. For example, hash a producer (for example, an instance) to a fixed shard based on the name to make sure the data writing and consumption in this shard are strictly ordered (when merging or splitting shards, a key can only appear in one shard at a time point). For more information, see Shard.

Request syntax

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-signaturemethod: hmac-sha1

 <Compressed log data in PB format>
```

KeyHash mode

Add x-log-hashkey in the header to determine which shard range the key belongs to. This parameter is optional. If left blank, Log Service automatically switches 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

Parameter name	Type	Required	Description
logstorename	string	Yes	The name of the Logstore where logs are to be written.

Request header

In the KeyHash mode, add the x-log-hashkey request header (see the preceding example). For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The PostLogstoreLogs API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

No response element after the successful request.

Detailed description

- You can use the PostLogstoreLogs API to write at most 4096 logs and the size of logs is at most 3 MB. The size of the value section in each log cannot be larger than 1 MB. The request fails and no logs are successfully written if any of the preceding conditions are not met.

- Log Service checks the format of the logs written by using the PostLogstoreLogs API (for more information about the log formats, see Core concepts). The request fails and no logs are successfully written if any log does not conform to the specification.

Error code

Besides the common error codes of Log Service APIs, the PostLogstoreLogs API may return the following special error codes.

HTTP status code	Error code	Error message	Description
400	PostBodyInvalid	Protobuffer content cannot be parsed.	The Protobuffer content cannot be parsed.
400	InvalidTimestamp	Invalid timestamps are in logs.	Invalid timestamps are in logs.
400	InvalidEncoding	Non-UTF8 characters are in logs.	Non-UTF8 characters are in logs.
400	InvalidKey	Invalid keys are in logs.	Invalid keys are in logs.
400	PostBodyTooLarge	Logs must be less than or equal to 3 MB and 4096 entries.	The number of logs must be no more than 4096 and the size of logs must be no more than 3 MB.
400	PostBodyUncompressError	Failed to decompress logs.	Failed to decompress logs.
499	PostBodyInvalid	The post data time is out of range.	The log time is out of the valid range [-7*24Hour, +15Min].
404	LogStoreNotExist	logstore {Name} does not exist.	The Logstore does not exist.

Note: The {name} in the preceding error message is replaced by a specific 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 of 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

Query the LogShipper task status.

Request syntax

```
GET
/logstores/{logstoreName}/shipper/{shipperName}/tasks?from=1448748198&to=1448948198&status=success&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

Parameter name	Type	Required	Description
logstoreName	string	Yes	The Logstore name,

			which is unique in the same project.
shipperName	string	Yes	The name of the log shipping rule, which is unique in the same Logstore.
from	integer	Yes	The start time of a LogShipper task.
to	integer	Yes	The end time of a LogShipper task.
status	string	No	The default value is empty, indicating that tasks of any status are returned. Currently, tasks in the successful, running, or failed status are returned.
offset	integer	No	The starting number of LogShipper tasks within a specified time range. The default value is 0.
size	integer	No	The number of LogShipper tasks within a specified time range. The default value is 100. The maximum value is 500.

Request header

The GetShipperStatus API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetShipperStatus API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

After the successful request, the response body contains a list of specified LogShipper 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	integer	The total number of tasks within a specified range.
statistics	json	The statistics of task status within a specified range. For more information, see the following table.
tasks	array	The details of a LogShipper task within a specified range. For more information, see the following table.

Statistics of task status

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

Task details

Name	Type	Description
id	string	The unique ID of a

		LogShipper task.
taskStatus	string	The LogShipper task status, which may be running, successful, and failed.
taskMessage	string	The error message appeared when a LogShipper task fails.
taskCreateTime	integer	The created time of a LogShipper task.
taskLastDataReceiveTime	integer	The time when the server receives the last log of a LogShipper task (the receipt time on the server, not the log time).
taskFinishTime	integer	The end time of a LogShipper task.

Error code

Besides the common error codes of Log Service APIs, the GetShipperStatus API may return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ShipperNotExist	shipper {logstoreName} does not exist
500	InternalServerError	internal server error
400	ParameterInvalid	start time must be earlier than end time
400	ParameterInvalid	only supports retrying tasks failed within 48 hours
400	ParameterInvalid	status only contains success/running/fail

Detailed description

You can only query LogShipper task status 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

Rerun failed LogShipper 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	Description
logstoreName	string	Yes	The Logstore name, which is unique in the same project.
shipperName	string	Yes	The name of the log shipping rule, which is unique in the same Logstore.

Request header

The RetryShipperTask API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The RetryShipperTask API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the RetryShipperTask API may return the following special error codes.

HTTP status code	Error code	Error message
404	ProjectNotExist	Project {ProjectName} does not exist
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ShipperNotExist	shipper {logstoreName} does not exist
500	InternalServerError	Specified Server Error Message
400	ParameterInvalid	Each time allows 10 task retries only

Detailed description

You can rerun at most 10 failed LogShipper tasks 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

Query logs in a Logstore of a specific project. You can also query the logs that meet the specific condition by specifying the relevant parameters.

When a log is written to the Logstore, the latency of querying this log by using Log Service query APIs (GetHistograms and GetLogs) varies according to the log type. Log Service classifies logs based on the log timestamp into the following two types:

> Real-time data: The time point in a log is the current time point on the server (-180 seconds, 900 seconds]. For example, if the log time is UTC 2014-09-25 12:03:00 and the time when the server receives the log is UTC 2014-09-25 12:05:00, the log is processed as the real-time data, which usually appears in normal scenarios.
>
> Historical data: The time point in a log is the current time point on the server (-7 x 86400 seconds, -180 seconds]. For example, if the log time is UTC 2014-09-25 12:00:00 and the time when the server receives the log is UTC 2014-09-25 12:05:00, the log is processed as the historical data, which usually appears in the supplementary data scenario.

The maximum latency between real-time data writing and query is 3 seconds. (data can be queried within one second in 99.9% cases).

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

Parameter name	Type	Required	Description
logstorename	string	Yes	The name of the Logstore where the log to be queried belongs.
type	string	Yes	The type of Logstore data to be queried. This parameter must be log in GetLogs API.
from	integer	Yes	The query start time (the number of seconds since 1970-1-1 00:00:00 UTC).
to	integer	Yes	The query end time (the number of seconds since 1970-1-1 00:00:00 UTC).
topic	string	No	The topic of the log to be queried.
query	string	No	The query expression. For more information about the query expression syntax, see Query syntax.
line	integer	No	The maximum number of logs returned from the request. The value range is 0–100 and the default value is 100.
offset	integer	No	The returned log start point of the request. The value can be 0 or a positive integer. The default value is 0.
reverse	boolean	No	Whether or not logs are returned in reverse order according to the log timestamp. true indicates reverse order and false indicates sequent order. The default

			value is false.

Request header

The GetLogs API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

For more information about the public response headers of Log Service APIs, see Public response header.

The response header has special elements to indicate whether or not the returned results of the request is complete. See the following specific response element formats.

Name	Type	Description
x-log-progress	string	The status of the query results. The two optional values Incomplete and Complete indicate whether or not the results are complete.
x-log-count	integer	The total number of logs in the current query results.

Response element

After the successful request, the response body contains the logs that meet the query conditions. The response results of this API may be incomplete when the log volume to be queried is large (T-level). The response body of GetLogs API is an array, and each element in this array is a log. The structure of each element in this array is as follows.

Name	Type	Description
__time__	integer	The log timestamp (the number of seconds since 1970-1-1 00:00:00 UTC).
__source__	string	The log source, which is specified when logs are written.
[content]	key-value pair	The original content of the log, which is organized in key-value pairs.

Detailed description

- The time interval defined by the request parameters from and to in this API follows the left-closed and right-opened principle, that is, the time interval includes the start time, but not the end time. If the from and to values are the same, the time interval is invalid and the function returns an error directly.
- Each call to this API must return results within a specified time, and each query can only scan a specified number of logs. The results returned from this request are incomplete if the log volume to be processed for this request is large (whether or not the results are complete is indicated by using the x-log-progress in the response header). At the same time, Log Service caches the query results within 15 minutes. If some query request results are the same as those in the cache, Log Service continues to scan the logs that are not in the cache for this request. To reduce the workload of merging multiple query results, Log Service merges the query results that are the same as those in the cache and the results newly scanned in this query, and then returns them to you. Therefore, Log Service allows you to call the API multiple times with the same parameter to obtain the final complete results. Log Service API cannot predict how many times the API must be called before obtaining the complete results because the log volume to be queried changes massively. Therefore, you must check the x-log-progress status in the returned results of each request to determine whether or not to continue the query. You must note that each call to this API consumes the same number of query CUs again.

Error code

Besides the common error codes of Log Service APIs, the GetLogs API may return the following special error codes.

HTTP status code	Error code	Error message	Description
404	LogStoreNotExist	logstore {Name} does not exist.	The Logstore does not exist.
400	InvalidTimeRange	request time range is invalid	The time interval of the request is invalid.
400	InvalidQueryString	query string is invalid	The query string of the request is invalid.
400	InvalidOffset	offset is invalid	The offset parameter of the request is invalid.
400	InvalidLine	line is invalid	The line parameter of the request is invalid.
400	InvalidReverse	Reverse value is invalid	The Reverse parameter value is

			invalid.
400	IndexConfigNotExist	logstore without index config	The Logstore does not enable the index.

Note: The {name} in the preceding error message is replaced by a specific Logstore name.

Example

Take a project named big-game in the region Hangzhou as an example. Query the logs whose topic is groupA in the app_log Logstore of the big-game project. The time interval for this query is 2014-09-01 00:00:00–2014-09-01 22:00:00. The keyword for this query is **error**. The query starts from the beginning of the time interval, 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
x-log-progress : Complete
x-log-count : 10000
x-log-processed-rows: 10000
x-log-elapsed-millisecond:5
{
"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 x-log-progress status is Complete, which indicates the log query is completed and the returned results are complete. For this request, two logs meet the query condition and are displayed as the values of logs. If the x-log-progress status is Incomplete in the response result, you must repeat the request to obtain the complete results.

GetHistograms

Query the log distribution in a Logstore of a specific project. You can also query the distribution of logs that meet the specific conditions by specifying the relevant parameters.

When a log is written to the Logstore, the latency of querying this log by using Log Service query APIs (GetHistograms and GetLogs) varies according to the log type. Log Service classifies logs based on the log timestamp into the following two types:

- Real-time data: The time point in a log is the current time point on the server (-180 seconds, 900 seconds]. For example, if the log time is UTC 2014-09-25 12:03:00 and the time when the server receives the log is UTC 2014-09-25 12:05:00, the log is processed as the real-time data, which usually appears in normal scenarios.
- Historical data: The time point in a log is the current time point on the server (-7 x 86400 seconds, -180 seconds]. For example, if the log time is UTC 2014-09-25 12:00:00 and the time when the server receives the log is UTC 2014-09-25 12:05:00, the log is processed as the historical data, which usually appears in the supplementary data scenario.

The latency between real-time data writing and query is 3 seconds.

Request syntax

```
GET
/logstores/<logstorename>?type=histogram&topic=<logtopic>&from=<starttime>&to=<endtime>&query=<querystring> 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

Parameter name	Type	Required	Description
logstorename	string	Yes	The name of the Logstore where the log to be queried belongs.
type	string	Yes	The type of Logstore data to be queried. This parameter must be histogram in GetHistograms API.
from	integer	Yes	The query start time (the number of seconds since 1970-1-1 00:00:00 UTC).
to	integer	Yes	The query end time (the number of seconds since 1970-1-1 00:00:00 UTC).
topic	string	No	The topic of the log to be queried.
query	string	No	The query expression. For more information about the query expression syntax, see Query syntax.

Request header

The GetHistograms API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetHistograms API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

After the successful request, the response body contains the distribution of logs that meet the query conditions on the timeline. The response results evenly divide the time range into several (1–60) subintervals and return the number of logs that meet the query conditions in each subinterval. Log Service must return results within a specified time to guarantee the timeliness. Therefore, each query can only scan a specified number of logs. The response results of this API may be incomplete when the log volume to be queried is large. A special response element is used to indicate whether or not the returned results of the request is complete. See the following specific response element formats.

Name	Type	Description
progress	string	The status of the query results. The two optional values Incomplete and Complete indicate whether or not the results are complete.
count	integer	The total number of logs in the current query results.
histograms	array	The distribution of the current query results in the subintervals. For more information about the structure, see the following table.

The structure of each element in the histograms array is as follows.

Name	Type	Description
from	integer	The start time for the subinterval (the number of seconds since 1970-1-1 00:00:00 UTC).
to	integer	The end time for the subinterval (the number of seconds since 1970-1-1 00:00:00 UTC).
count	integer	The number of logs that meet the query conditions for this subinterval in the current query results.
progress	string	Whether or not the current query results in this subinterval are complete. Optional values: Incomplete and Complete.

Detailed description

- All the time intervals in this API, whether the time intervals defined by the request parameters from and to, or subintervals in the returned results, follow the left-closed and right-opened principle, that is, the time interval includes the start time, but not the end time. If the from and to values are the same, the time interval is invalid and the function returns an error directly.
- The subinterval division method in the response of this API is consistent and unchanging. The subinterval division in the response does not change if the time interval of your request does not change.
- Each call to this API must return results within a specified time, and each query can only scan a specified number of logs. The results returned from this request are incomplete if the log volume to be processed for this request is large (whether or not the results are complete is indicated by using the progress in the returned results). At the same time, Log Service caches the query results within 15 minutes. If some query request results are the same as those in the cache, Log Service continues to scan the logs that are not in the cache for this request. To reduce the workload of merging multiple query results, Log Service merges the query results that are the same as those in the cache and the results newly scanned in this query, and then returns them to you. Therefore, Log Service allows you to call the API multiple times with the same parameter to obtain the final complete results. Log Service API cannot predict how many times the API must be called before obtaining the complete results because the log volume to be queried changes massively. Therefore, you must check the progress value in the returned results of each request to determine whether or not to continue the query. You must note that each call to this API consumes the same number of query CUs again.

Error code

Besides the common error codes of Log Service APIs, the GetHistograms API may return the following special error codes.

HTTP status code	Error code	Error message	Description
404	LogStoreNotExist	logstore {Name} does not exist.	The Logstore does not exist.
400	InvalidTimeRange	request time range is invalid.	The time interval of the request is invalid.
400	InvalidQueryString	query string is invalid.	The query string of the request is invalid.

Note: The {name} in the preceding error message is replaced by a specific Logstore name.

Example

Take a project named big-game in the region Hangzhou as an example. Query the distribution of logs whose topic is groupA in the app_log Logstore of the big-game project. The time interval for this query is 2014-09-01 00:00:00–2014-09-01 22:00:00. The keyword for this query 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, Log Service divides the entire Histogram into two equal time intervals: [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). The returned results of the first query are incomplete because your log volume to be queried is large. The response results indicate that three logs meet the query conditions, but the overall results are incomplete. Only the results in the time interval [2014-09-01 00:00:00, 2014-09-01 11:00:00) are complete, with two logs meeting the query conditions. However, the results in the other time interval are incomplete, with one log meeting the query conditions. In this situation, to obtain the complete results, you must call the preceding request example multiple times until the progress value in the response changes to Complete as follows.

```
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

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name, which is unique in the same project.
groupType	string	No	The machine group type, which is empty by default.
machineIdentifyType	string	Yes	The machine identification type, including IP and user-defined identity.
groupAttribute	object	Yes	The machine group attribute, which is

			empty by default.
machineList	array	Yes	The specific machine identification, which can be an IP address or user-defined identity.

groupAttribute description

Attribute name	Type	Required	Description
groupTopic	string	No	The topic of a machine group, which is empty by default.
externalName	string	No	The external identification that the machine group depends, which is empty by default.

Request header

The CreateMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The CreateMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the CreateMachineGroup API may return the following special error codes.

HTTP status code	Error code	Error message
400	MachineGroupAlreadyExist	group {GroupName} already exists
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

Delete a machine group and the Logtail configuration applied to this machine group.

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

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name.

Request header

The DeleteMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The DeleteMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does 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

Update the machine group information. If the machine group has applied a configuration, the configuration is automatically added or removed when machines are added to or removed 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

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name.

Body parameters

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name, which is unique in the same project.
groupType	string	No	The machine group type, which is empty by default.
machineIdentifyType	string	Yes	The machine identification type, including IP and user-defined identity.
groupAttribute	object	Yes	The machine group attribute, which is empty by default.
machineList	array	Yes	The specific machine identification, which can be an IP address or user-defined identity.

groupAttribute description

Attribute name	Type	Required	Description
groupTopic	string	No	The topic of a machine group, which is empty by default.
externalName	string	No	The external identification that the machine group depends, which is empty by default.

Request header

The UpdateMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The UpdateMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the UpdateMachineGroup API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does 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

Parameter name	Type	Required	Description
offset	integer	No	The starting position of the returned records. The default value is 0.
size	integer	No	The maximum

			number of entries returned on each page. The default value is 500 (maximum).
groupName	string	No	The group machine name used for filtering (partial matching is supported).

Request header

The ListMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The ListMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

After the successful request, the response body contains a list of all machine groups in a specific project. The specific formats are as follows.

Name	Type	Description
count	integer	The number of returned machine groups.
total	integer	The total number of returned machine groups.
machinegroups	json array	The name list of returned machine groups.

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

Error code

Besides the common error codes of Log Service APIs, the ListMachineGroup API may return the following special error codes.

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

View 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

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name.

Request header

The GetMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

Attribute name	Type	Description
groupName	string	The machine group name, which is unique in the same project.
groupType	string	The machine group type (empty or Armory), which is empty by default.
machineIdentifyType	string	The machine identification type, including IP and user-defined identity.
groupAttribute	json object	The machine group attribute, which is empty by default.
machineList	json array	The specific machine identification, which can be an IP address or user-defined identity.
createTime	integer	The created time of the machine group.
lastModifyTime	integer	The last updated time of the machine group.

groupAttribute description

Attribute name	Type	Required	Description
groupTopic	string	No	The topic of a machine group, which is generally not configured.
externalName	string	No	The external system (Armory) identification that the machine group depends.

```
{
"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

Besides the common error codes of Log Service APIs, the GetMachineGroup API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does 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

Apply a configuration to a machine group.

Example:

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

Request parameters

Parameter name	Type	Required	Description
GroupName	string	Yes	The machine group name.
ConfigName	string	Yes	The Logtail configuration name.

Request header

The ApplyConfigToMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The ApplyConfigToMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the ApplyConfigToMachineGroup API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
404	ConfigNotExist	config {ConfigName} does 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

Remove a configuration from a machine group.

Example:

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

Request parameters

Parameter name	Type	Required	Description
GroupName	string	Yes	The machine group name.
ConfigName	string	Yes	The Logtail configuration name.

Request header

The RemoveConfigFromMachineGroup API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The RemoveConfigFromMachineGroup API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the RemoveConfigFromMachineGroup API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
404	ConfigNotExist	config {ConfigName} does 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

Obtain the status of your machine that is in the machine group and connected to the server.

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

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name.
offset	integer	No	The starting position of the returned records. The default value is 0.
size	integer	No	The maximum number of entries returned on each page. The default value is 500 (maximum).

Request header

The ListMachines API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The ListMachines API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

Name	Type	Description
count	integer	The number of returned machines.
total	integer	The total number of machines.
machines	json array	The name list of returned machines.

Machine description

Name	Type	Description
ip	string	The IP address of the machine.
machine-uniqueid	string	The DMI UUID of the machine.
userdefined-id	string	The user-defined identity 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

Besides the common error codes of Log Service APIs, the ListMachines API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does not exist
500	InternalServerError	internal server error

Detailed description

This API only obtains the list of machines that are normally connected to the server.

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

Obtain the name of the configuration applied to a 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

Parameter name	Type	Required	Description
groupName	string	Yes	The machine group name.

Request header

The GetAppliedConfigs API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetAppliedConfigs API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

After the successful request, the response body contains a list of all machines in a specific machine group. The specific formats are as follows.

Name	Type	Description
count	integer	The number of returned configurations.
configs	string array	The name list of returned configurations.

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

Error code

Besides the common error codes of Log Service APIs, the GetAppliedConfigs API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does 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

Create a Logtail configuration in 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

Parameter	Type	Required	Description	

name			
configName	string	Yes	The Logtail configuration name, which is unique in the same project.
inputType	string	Yes	The input type. Currently, only file is supported.
inputDetail	json	Yes	See the descriptions in the following table.
outputType	string	Yes	The output type. Currently, only LogService is supported.
outputDetail	json	Yes	See the descriptions in the following table.
logSample	string	No	The log sample of the Logtail configuration. The log size cannot exceed 1,000 bytes.

inputDetail contents

Attribute name	Type	Required	Description
logType	string	Yes	The log type. Currently, only common_reg_log is supported.
logPath	string	Yes	The parent directory where the log resides. For example, /var/logs/.
filePattern	string	Yes	The pattern of a log file. For example, access*.log.
localStorage	boolean	Yes	Whether or not to activate the local cache. Logs of 1 GB can be cached locally when the link to Log Service is 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, which is used to match with logs composed of multiple lines.
regex	string	Yes	The regular expression used for extracting logs.
key	array	Yes	The key generated after logs are extracted.
filterKey	array	Yes	The key used for filtering logs. The log meets the 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 the same as that of filterKey.
topicFormat	string	No	The topic generation mode. The four supported modes are as follows: - Use a part of the log file path as the topic. For example, /var/log/(.*).log. - none indicates the topic is empty.

			- default indicates to use the log file path as the topic. - group_topic indicates to use the topic attribute of the machine group that applies this configuration as the topic.
preserve	boolean	No	true indicates that the monitored directory never times out. false indicates that the timeout for monitored directory is 30 minutes. The default value is true.
preserveDepth	integer	No	If preserve is set to false, specify the depth of the directories with no monitoring timeout. The maximum depth is 3.
fileEncoding	string	No	Two types are supported: utf8 and gbk.

outputDetail content

Attribute name	Type	Required	Description
logstoreName	string	Yes	The Logstore name.

Request header

The CreateConfig API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The CreateConfig API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the CreateConfig API may return the following special error codes.

HTTP status code	Error code	Error message
400	ConfigAlreadyExist	config {Configname} already exists
400	InvalidParameter	invalid config resource json
500	InternalServerError	internal server error

Detailed description

The configuration fails to be created if an error occurs during the creation, for example, the configuration already exists, the format is incorrect, the required parameters are missing, or the quota is exceeded.

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

List all the configurations in 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

Parameter name	Type	Required	Description
offset(optional)	integer	No	The starting position of the returned records. The default value is 0.
size(optional)	integer	No	The maximum number of entries returned on each page. The default value is 500 (maximum).

Request header

The ListConfig API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The ListConfig API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The response body contains a list of all configurations in a specific project. The specific formats are as follows.

Name	Type	Description
count	integer	The number of returned configurations.
total	integer	The total number of configurations in Log Service.
configs	string array	The name list of returned configurations.

Error code

Besides the common error codes of Log Service APIs, the ListConfig API may return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	config {Configname} does not exist
500	InternalServerError	internal server error

Detailed description

None.

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

List the machines that apply the configuration.

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

Parameter name	Type	Required	Description
ConfigName	string	Yes	The configuration name.

Request header

The GetAppliedMachineGroups API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetAppliedMachineGroups API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

After the successful request, the response body contains a list of all machines in a specific machine group. The specific formats are as follows.

Name	Type	Description
count	integer	The number of returned machine groups.
machinegroups	string array	The name list of returned machine groups.

```
{
"count":2,
"machinegroups":
["group1","group2"]
}
```

Error code

Besides the common error codes of Log Service APIs, the GetAppliedMachineGroups API may return the following special error codes.

HTTP status code	Error code	Error message
404	GroupNotExist	group {GroupName} does 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

Obtain the 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

Parameter name	Type	Required	Description
ConfigName	string	Yes	The Logtail configuration name.

Request header

The GetConfig API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The GetConfig API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

Attribute name	Type	Description
configName	string	The Logtail configuration name, which is unique in the same project.
inputType	string	The input type. Currently, only file is supported.

inputDetail	json	See the descriptions in the following table.
outputType	string	The output type. Currently, only LogService is supported.
outputDetail	json	See the descriptions in the following table.
createTime	integer	The created time of the configuration.
lastModifyTime	integer	The resource updated time in Log Service.

inputDetail contents

Attribute name	Type	Description
logType	string	The log type. Currently, only common_reg_log is supported.
logPath	string	The parent directory where the log resides. For example, /var/logs/.
filePattern	string	The pattern of a log file. For example, access*.log.
localStorage	boolean	Whether or not to activate the local cache. Logs of 1 GB can be cached locally when the link to Log Service is disconnected.
timeFormat	string	The format of log time. For example, %Y/%m/%d %H:%M:%S.
logBeginRegex	string	The characteristics (regular expression) of the first log line, which is used to match with logs composed of multiple lines.
regex	string	The regular expression used for extracting logs.
key	array	The key generated after logs are extracted.
filterKey	array	The key used for filtering logs. The log meets the requirements only when the key 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 the same as that of filterKey.
topicFormat	string	Use a part of the log file path as the topic. For example, /var/log/(.*).log. The default value is none, which indicates the topic is empty.
preserve	boolean	true indicates that the monitored directory never times out. false indicates that the timeout for monitored directory is 30 minutes. The default value is true.
preserveDepth	integer	If preserve is set to false, specify the depth of the directories with no monitoring timeout. The maximum depth is 3.
fileEncoding	string	The encoding format of the log file, which supports utf8 and gbk.

outputDetail content

Attribute name	Type	Required	Description
endpoint	string	Yes	The access address of the region where the project resides.
logstoreName	string	Yes	The Logstore name.

Error code

Besides the common error codes of Log Service APIs, the GetConfig API may return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	Config {Configname} does not exist
500	InternalServerError	Specified Server Error Message

Detailed description

None.

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

Delete a specific configuration. If the configuration has been applied to the machine group, logs cannot be collected based on this configuration.

```
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

Parameter name	Type	Required	Description
ConfigName	string	Yes	The configuration name.

Request header

The DeleteConfig API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The DeleteConfig API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the DeleteConfig API may return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	config {Configname} does not 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

Update the configuration. If the configuration is applied to a machine group, the corresponding machines are also 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",
"logBeginRegex": ".*",
"regex": "(\w+)(\s+)",
"key" :["key1", "key2"],
"filterKey":["key1"],
"filterRegex":["regex1"],
"topicFormat": "none"
},
"outputType": "LogService",
"outputDetail":
{
"logstoreName": "perfcounter"
}
}
```

Request parameters

Attribute name	Type	Required	Description
configName	string	Yes	The Logtail configuration name, which is unique in the same project.
inputType	string	Yes	The input type. Currently, only file is supported.
inputDetail	json	Yes	See the descriptions in the following table.
outputType	string	Yes	The output type. Currently, only LogService is supported.
outputDetail	json	Yes	See the descriptions in the following table.

inputDetail contents

Attribute name	Type	Required	Description
logType	string	Yes	The log type. Currently, only common_reg_log is supported.
logPath	string	Yes	The parent directory where the log resides. For example, /var/logs/.
filePattern	string	Yes	The pattern of a log file. For example, access*.log.
localStorage	boolean	Yes	Whether or not to activate the local cache. Logs of 1 GB can be cached locally when the link to Log Service is

			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, which is used to match with logs composed of multiple lines.
regex	string	Yes	The regular expression used for extracting logs.
key	array	Yes	The key generated after logs are extracted.
filterKey	array	Yes	The key used for filtering logs. The log meets the 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 the same as that of filterKey.
topicFormat	string	No	Use a part of the log file path as the topic. For example, /var/log/(.*).log. The default value is none, which indicates the topic is empty.
preserve	boolean	No	true indicates that the monitored directory never times out. false indicates that the timeout for monitored directory is 30 minutes. The default value is true.

preserveDepth	integer	No	If preserve is set to false, specify the depth of the directories with no monitoring timeout. The maximum depth is 3.
fileEncoding	string	No	Two types are supported: utf8 and gbk. The default value is utf8.

outputDetail content

Attribute name	Type	Required	Description
logstoreName	string	Yes	The Logstore name.

Request header

The UpdateConfig API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

The UpdateConfig API does not have a special response header. For more information about the public response headers of Log Service APIs, see Public response header.

Response element

The returned HTTP status code is 200.

Error code

Besides the common error codes of Log Service APIs, the UpdateConfig API may return the following special error codes.

HTTP status code	Error code	Error message
404	ConfigNotExist	config {Configname} does not exist
400	InvalidParameter	invalid config resource json
400	BadRequest	config resource configname does not match with request
500	InternalServerError	internal server error

Detailed description

The configuration fails to be created if an error occurs during the creation, for example, the format is incorrect, the required parameters are missing, or the quota is exceeded.

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

Access Log Service resources of your primary account as a RAM user after RAM authorization

The projects, Logstores, configurations, and machine groups you create are your own resources. By default, you have the full operation permissions to 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 Resource Access Management (RAM) user, the RAM user cannot perform operations on resources of the primary account after being created. You must grant permissions to the RAM user to perform operations on resources of the primary account by using RAM authorization.

> **Note:** Before using RAM to grant a RAM user the permissions to access Log Service resources of a primary account, make sure that you have carefully read Create a RAM user and RAM introduction.

Three authorization policies for Log Service are available in the RAM console.

AliyunLogFullAccess

This policy is used to grant a RAM user the full access permission to 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 RAM user the read-only permission to 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 specific Logstore in the console

This policy is used to grant a RAM user the read-only permission to the resources of a primary account’ s specific Logstore. After the authorization, the RAM user can query logs, extract logs, and view Logstore list in the console. The authorization policy is described as follows:

```
 {
"Version": "1",
"Statement": [
{
"Action": ["log:ListProject", "log:ListLogStores"],
"Resource": ["acs:log:*:*:project/<specific project name>/*"],
"Effect": "Allow"
},
{
"Action": ["log:Get*"],
"Resource": ["acs:log:*:*:project/<specific project name>/logstore/<specific Logstore name>"],
"Effect": "Allow"
}
]
}
```

To not grant a RAM user the permissions to access Log Service resources of a primary account, skip this section. Skipping this section does not affect your understanding and usage of Log Service.

For more information, see:

- Types of Log Service resources that can be authorized in RAM
- Actions in RAM that can be performed on Log Service resources
- Authentication rules used when a RAM user accesses the resources of a primary account by using Log Service APIs

Resource list

Types of Log Service resources that can be authorized in RAM

The types of resources that can be authorized in Resource Access Management (RAM) and the description methods are as follows.

Resource type	Description method 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/${projectName}/logtailconfig/${logtail

	config}
	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/${logstoreName}/consumergroup/*
Project/SavedSearch	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/savedsearch/${savedSearchName}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/savedsearch/*
Project/Dashboard	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/dashboard/${dashboardName}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/dashboard/*
Project/Alarm	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/alert/${alarmName}
	acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/alert/*
Generic mode	acs:log:${regionName}:${projectOwnerAliUid}:*
	acs:log:*:${projectOwnerAliUid}:*

Note: A hierarchical relationship is in the Log Service resources. The project is a top-level resource. Logstore, configuration, and machine group are at the same level and sub-resources of the project. Log shipping rule and consumer group are sub-resources of the Logstore.

Wherein:

- ${regionName} indicates the name of a region.
- ${projectOwnerAliUid} indicates your 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 configuration.
- ${machineGroupName} indicates the name of a machine group.
- ${shipperName} indicates the name of a log shipping rule.
- ${consumerGroupName} indicates the name of a consumer group.
- ${savedSearchName} indicates the name of a saved search.
- ${dashboardName} indicates the name of a dashboard.
- ${alarmName} indicates the name of an alarm rule.

Action list

Actions in RAM that can be performed on Log Service resources

In Resource Access Management (RAM), you can perform the following actions on Log Service resources. 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

Authentication rules used when a RAM user accesses the resources of a primary account by using Log Service APIs

When a Resource Access Management (RAM) user accesses the resources of a primary account by using Log Service APIs, Log Service backend performs RAM permission inspection to make sure the resource owner has granted relevant permissions to the caller.

Different Log Service APIs determine the resources whose permissions must be checked according to the involved resources and the meanings of the API. The authentication rules for each API are as follows.

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}**

Configuration

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/*

Machine group

Action	Resource
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}**

Interactive APIs for configuration and machine group

Action	Resource

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

Overview

Access Log Service resources of another account by using STS

The projects, Logstores, configurations, and machine groups you create are your own resources. By default, you have the full operation permissions to your resources, and can use all APIs in this document to perform operations on your resources.

To grant another account the permissions to access your resources, you must use Security Token Service (STS) to obtain the temporary AccessKey/token to call specific operations. Before reading the following instructions, see the STS product document.

Assume that user A creates projects, Logstores, and other resources in Log Service, and user B wants to call an API to access these resources, the procedure is as follows.

User A

Create a role

User A creates a role for the trusted account B in the Resource Access Management (RAM) console or by using API. The role details are as follows:

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

Grant permissions to the created role

After a role is created, user A must grant specific operation permissions to the role.

The permissions required for data writing only are described as follows:

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

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

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

Configure the resource as follows:

- The preceding two types of resources authorize the role to access all the projects and Logstores of a specific user.
- To authorize the role to access a specific project: acs:log:*:{projectOwnerAliUid}:project/*
- To authorize the role to access a specific Logstore: acs:log:*:{projectOwnerAliUid}:project/{projectName}/logstore/{logstoreName}/*

For complete resource description, see Log Service RAM resources.

User B

Create and authorize a RAM user

Create a RAM user and grant the AssumeRole permission to the created RAM user in the RAM console or by using APIs/SDKs.

Call STS interface to obtain the temporary AccessKey/token

For more information, see STS SDK usage instructions.

Call a Log Service interface

For more information, see Log Service SDK usage instructions.

Sample code

The sample code, which is based on Java SDK, is applicable to the case where user B writes data to projects of user A by using STS.

Code link

Common resources

Data model

For easy understanding and use of Log Service, see the following basic concepts first.

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 users for lower access latency and better user experience. Currently, Alibaba Cloud has multiple regions throughout the country.

Project

The project is a basic 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 an application.

Logstore

The Logstore is a unit in Log Service to collect, store, and consume logs. Each Logstore belongs to a project, and each project can create multiple Logstores. You can create multiple Logstores for a project according to your actual needs. Typically, an independent Logstore is created for each type of logs in an application. For example, you have a game application **big-game**, and three types of logs are on the server: operation_log, application_log, and access_log. You can first create a project named **big-game**, and then create three Logstores in this project for these three types of logs to collect, store, and consume logs respectively.

Log

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

- **Topic:** A user-defined field used to mark multiple logs. For example, access logs can be marked according to sites. By default, this field is an empty string, which is also a valid topic.
- **Time:** A reserved field in the log used to indicate the log generation time (the number of seconds since 1970-1-1 00:00:00 UTC). Generally this field is generated directly based on the time in the log.
- **Content:** A field used to record the specific log content. The log content is composed of one or more content items, and each content item is a key-value pair.
- **Source:** A field used to indicate the source of the log. For example, the IP address of the machine where the log is generated. By default, this field is empty.

Log Service has different requirements on values of different fields as follows.

Data field	Requirement
time	An integer in the standard UNIX time format. The unit is in seconds.
topic	A UTF-8 encoded string up to 128 bytes.
source	A UTF-8 encoded string up to 128 bytes.
content	One or more key-value pairs. The key is a UTF-8 encoded string up to 128 bytes, which can contain letters, underscores (_), and numbers, but cannot start with a number. The value is a UTF-8 encoded string up to 1024*1024 bytes.

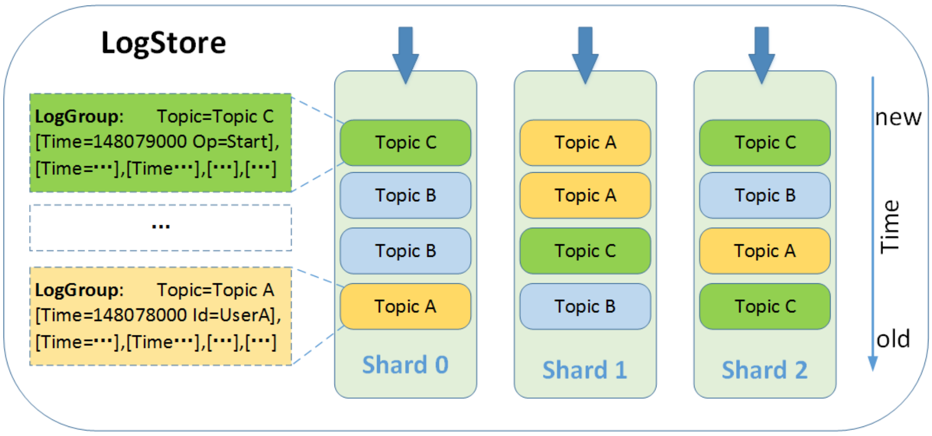
The key in the content cannot use any of the following keywords: __time__, __source__, __topic__, __partition_time__, _extract_others_, and __extract_others__.

Topic

Logs in a Logstore can be classified by log topics. You can specify the topic when writing logs. For example, as a platform user, you can use your user ID as the log topic when writing logs. To not classify the logs in a Logstore, use the same topic for all of the logs.

Note: An empty string is a valid log topic and is the default log topic.

The relationship among Logstores, log topics, and logs is as follows.



Various log formats are used in actual usage scenarios. For better understanding, the following example describes how to map an original Nginx access log to the Log Service log data model. Assume that the IP address of your Nginx server is 10.249.201.117. An original log of this server is as follows.

```
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 Service log data model as follows.

Data field	Content	Description
topic	“”	Use the default value (empty string).
time	1330589527	The precise log generation time (in seconds), which is converted from the timestamp of the original log.
source	“10.249.201.117”	Use the IP address of the server as the log source.
content	key-value pair	Specific log content.

You can decide how to extract the original log contents and combine them into key-value pairs. The following table is shown as an example.

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 logs.

Log group

A group of logs.

Log group list

A collection of log groups used to return results.

Encoding method

Currently, the system supports the following content encoding method. 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 Protocol Buffer (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: PB does not require the key-value pair to be unique. You must avoid such situation.

Otherwise, the behavior is undefined.

Data encoding method

Protocol Buffer is a structured data interchange format developed by Google. It is widely used in many internal and external services of Google. Currently, Log Service uses Protocol Buffer format as the standard log writing format. You must serialize the original log data into Protocol Buffer data streams before writing logs to Log Service by using APIs.

```
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:

- PB does not require the key-value pair to be unique. You must avoid such situation. Otherwise, the behavior is undefined.
- For more information about Protocol Buffer format, see GitHub.
- For more information about the API for writing logs to Log Service, see PostLogstoreLogs.

Logstore

The Logstore is a data storage unit. By default, you can create at most 10 Logstores in a project. The Logstore name must be unique in the same project. The Logstore is the endpoint for all logs and you can write or read logs to or from the Logstore.

The Logstore naming rules are as follows:

- The name can only contain lowercase letters, numbers, hyphens (-), and underscores (_).
- The name must begin and end with a lowercase letter or number.
- The name must be 3–63 bytes long.

Example of the complete resource:

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

Parameter definitions

Parameter name	Type	Required	Description
logstoreName	string	Yes	The Logstore name, which must be unique in the same project.
ttl	integer	Yes	The Time to Live (TTL) of log data. The unit is in days and the minimum value is 1 day.
shardCount	integer	Yes	The log data service unit.
createTime(OutputOnly)	integer	No	The time when the resource is created in Log Service (output only).
lastModifyTime(OutputOnly)	integer	No	The time when the resource is updated in Log Service (output only).

Shard

The shard is the basic unit to read and write logs in each Logstore. You can specify the number of shards in each Logstore. Each shard has certain service capacities:

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

To read data from a shard, you must specify the corresponding shard. To write data to a shard, you can use Server Load Balancer. By using Server Load Balancer, data is automatically written to shards based on the system load in the backend, which guarantees the high availability of log writing.

Example of the complete resource

Parameter name	Type	Required	Description
shardID	integer	Yes	The unique ID of a shard in the Logstore, which is automatically generated by the system.

Logtail configuration

By default, you can create at most 100 Logtail configurations for a project. The configuration name must be unique in the same project.

You can use the configuration to specify the location, method, and parameters for log collection.

The configuration naming rules are as follows:

- The name can only contain lowercase letters, numbers, hyphens (-), and underscores (_).
- The name must begin and end with a lowercase letter or number.
- The name must be 2–128 bytes long.

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	Description
configName	string	Yes	The Logtail configuration name, which is unique in the same project.
inputType	string	Yes	The input type. Currently, only file is supported.
inputDetail	json	Yes	See the descriptions in the following table.
outputType	string	Yes	The output type. Currently, only LogService is supported.
outputDetail	string	Yes	See the descriptions in the following table.
createTime(output-only)	integer	No	The created time of the configuration.
lastModifyTime (output-only)	integer	No	The time when the resource is updated in Log Service.

inputDetail contents

Attribute name	Type	Required	Description
logType	string	Yes	The log type. Currently, only common_reg_log is supported.
logPath	string	Yes	The parent directory where the log resides. For example, /var/logs/.
filePattern	string	Yes	The pattern of a log file. For example, access*.log.
localStorage	boolean	Yes	Whether or not to activate the local cache. Logs of 1 GB can be cached locally when the link to Log Service is 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, which is used to match with logs composed of multiple lines.
regex	string	Yes	The regular expression used for extracting logs.
key	array	Yes	The key generated after logs are extracted.
filterKey	array	Yes	The key used for filtering logs. The log meets the 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 the same as that of filterKey.
topicFormat	string	No	Use a part of the log file path as the topic. For example, /var/log/(.*).log. The default value is none, which indicates the topic is empty.
preserve	boolean	No	true indicates that the monitored directory never times out. false indicates that the timeout for monitored directory is 30 minutes. The default value is true.
preserveDepth	integer	No	If preserve is set to false, specify the depth of the directories with no monitoring timeout. The maximum depth is 3.

outputDetail content

Attribute name	Type	Required	Description
logstoreName	string	Yes	The Logstore name.

Machine group

Machine

After a machine with Logtail installed is started normally, the machine is automatically associated with the current user based on the user information in the Logtail configuration. Currently, the machine can be identified in the following three ways:

- IP: The IP address corresponding to the hostname. This is the easiest way to understand, but the IP address may be duplicated in environments such as Virtual Private Cloud (VPC).

- UUID (machine-uniqueid): The UUID in DMI devices. For more information, see RFC4122.
- Userdefined-id: You can customize the 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 of the machine identification, which is uploaded by Logtail.
userdefined-id	string	The user-defined machine identification, which is uploaded by Logtail.
lastHeartbeatTime(output-only)	integer	The last heartbeat time of the machine (the number of seconds since the epoch time).

Machine group

You can identify your machine group in a project by using IP address or user-defined identity. The IP address is more easily identified while the user-defined identity can solve the problem of identical IP address in the VPC environment. You can select either of the two machine identification methods.

Machine group naming rules:

- The name can only contain lowercase letters, numbers, hyphens (-), and underscores (_).
- The name must begin and end with a lowercase letter or number.
- The name must be 2–128 bytes long.

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	Description
groupName	string	Yes	The machine group name, which is unique in the same project.
groupType	string	No	The machine group type, which is empty by default.
machineIdentifyType	string	Yes	The machine identification type, including IP and user-defined identity.
groupAttribute	object	Yes	The machine group attribute, which is empty by default.
machineList	array	Yes	The specific machine identification, which can be an IP address or user-defined identity.
createTime(output-only)	integer	No	The created time of the resource.
lastModifyTime(output-only)	integer	No	The time when the resource is updated in Log Service.

groupAttribute description

Attribute name	Type	Required	Description
groupTopic	string	No	The topic of a machine group, which is empty by default.
externalName	string	No	The external identification that the machine group depends, which is empty by default.

Project interface

GetProjectLogs

Count all the logs in a project.

Request syntax

```
GET /logs/?query=SELECT * FROM sls_operation_log where __line__ = 'abc' and __date__ >'2017-09-01 00:00:00' and
__date__ < '2017-09-02 00:00:00&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
```

Request parameters

Parameter name	Type	Required	Description
query	string	Yes	The SQL query condition.

Request header

The GetProjectLogs API does not have a special request header. For more information about the public request headers of Log Service APIs, see Public request header.

Response header

For more information about the public response headers of Log Service APIs, see Public response header.

The response header has a special element to indicate whether or not the returned results of the request is complete. See the following specific response element formats.

Name	Type	Description
x-log-progress	string	The status of the query results. The two optional values Incomplete and Complete indicate whether or not the results are complete.
x-log-count	integer	The total number of logs in the current query results.
x-log-processed-rows	integer	The number of rows processed in this calculation.
x-log-elapsed-millisecond	integer	The time (in milliseconds) spent in this calculation.

Response element

After the successful request, the response body contains the computing results. The response body of GetProjectLogs is an array, and each element in the array is a log.

The element formats are as follows.

Name	Type	Description
__time__	integer	The timestamp of the log (the number of seconds since 1970-1-1 00:00:00 UTC).
__source__	string	The source of the log, which is specified when writing logs.
[content]	key-value pair	The original content of the log, which is organized in the key-value pair.

Detailed description

- The query of this API is a standard SQL query statement.
- Specify the project you want to query in the domain name of the request.
- Specify the Logstore you want to query in the FROM condition of the query statement. Logstore is equivalent to the table in SQL.
- You must specify the time range you want to query in the SQL query condition. The time range is specified by __date__ (timestamp type) or __time__ (integer type, the unit is in UNIX time).

- Each call to this API must return results within a specified time, and each query can only scan a specified number of logs. The results returned from this request are incomplete if the log volume to be processed for this request is large (whether or not the results are complete is indicated by using the x-log-progress in the response header). At the same time, Log Service caches the query results within 15 minutes. If some query request results are the same as those in the cache, Log Service continues to scan the logs that are not in the cache for this request. To reduce the workload of merging multiple query results, Log Service merges the query results that are the same as those in the cache and the results newly scanned in this query, and then returns them to you. Therefore, Log Service allows you to call the API multiple times with the same parameter to obtain the final complete results. Log Service API cannot predict how many times the API must be called before obtaining the complete results because the log volume to be queried changes massively. Therefore, you must check the x-log-progress status in the returned results of each request to determine whether or not to continue the query. You must note that each call to this API consumes the same number of query CUs again.

Error code

Besides the common error codes of Log Service APIs, the GetProjectLogs API may return the following special error codes.

HTTP status code	Error code	Error message	Description
400	ParameterInvalid	parameter is invalid	The request parameter is invalid. For more information, see the detailed error message.

Example

Take a project named big-game in the region Hangzhou as an example. Query the logs whose topic is groupA in the app_log Logstore of the big-game project. The time interval for this query is 2014-09-01 00:00:00–2014-09-01 22:00:00. The keyword for this query is **error**. The query starts from the beginning of the time interval, and a maximum of 20 logs are returned.

Request example

```
GET /logs/?query=SELECT * FROM sls_operation_log where __line__ = 'abc' and __date__ >'2017-09-01 00:00:00' and __date__ < '2017-09-02 00:00:00&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
x-log-progress : Complete
x-log-count : 10000
x-log-processed-rows: 10000
x-log-elapsed-millisecond:5

{
"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 x-log-progress status is Complete, which indicates the log query is completed and the returned results are complete. For this request, two logs meet the query condition and are displayed as the values of logs. If the x-log-progress status is Incomplete in the response result, you must repeat the request to obtain the complete results.

SDK Reference

Introduction

Overview

To allow developers to use Log Service more efficiently, Log Service provides software development kits (SDKs) in multiple languages (Java, .NET, Python, PHP, and C). Select to use an appropriate version as per your needs.

Log Service SDKs are implemented based on Log Service APIs and provide the same capabilities as Log Service APIs. For more information about the Log Service APIs, see API reference.

Similar to Log Service APIs, you must have an enabled Alibaba Cloud AccessKey (consisting of AccessKey ID and AccessKey Secret) to use Log Service SDKs. For more information, see AccessKey.

To use Log Service SDKs, you must know the service endpoint of Log Service in each Alibaba Cloud region. For how to specify the root endpoint in an SDK, see SDK configurations.

Though the implementation details of Log Service SDKs vary with different languages, the SDKs can be considered as Log Service APIs encapsulated in different languages and basically implement the same functions as follows.

- Unified encapsulation of the Log Service APIs, removing your need to build specific API requests and parse responses. The interfaces in various languages are similar, facilitating your switch between different languages.
- Digital signature logic for the Log Service APIs, greatly reducing the complexity of using APIs as you can ignore details of the API signature logic.
- Encapsulation of logs collected to Log Service in the Protocol Buffer format, allowing you to write logs without caring about the details of Protocol Buffer format.
- Implementation of the compression method defined in the Log Service APIs, removing the need to focus on the compression details. SDKs in some languages allow you to specify whether or not to write logs in the compression mode. (By default, the compression mode is used.)
- Unified error handling method, allowing you to handle request exceptions in the method that languages are familiar with.
- Currently, SDKs in all languages only support synchronous requests.

The download addresses, usage instructions, and complete programming references of SDKs in different languages are as follows.

SDK language	Relevant document	Source code
Java	Quick start and Interface reference	GitHub
.NET	Quick start and Interface reference	GitHub

PHP	Quick start and Interface reference	GitHub
Node.js		GitHub
Python	Quick start, Interface reference, and README	GitHub
C	Usage instructions	GitHub
GO	Usage instructions	GitHub
iOS	Usage instructions	GitHub
Android	Usage instructions	GitHub

Configurations

Like using APIs to interact with Log Service, you must also specify basic configurations when using SDKs. Currently, SDKs in all languages define a client class as the endpoint class. Basic configurations are specified when the endpoint class is built and include the following items:

- Service endpoint: Specify the service endpoint that the client must access.
- Alibaba Cloud AccessKey (consisting of AccessKey ID and AccessKey Secret): Specify the AccessKey used by the client to access Log Service.

For how to use the two configuration items, see the following sections.

Service endpoint

When using SDKs, you must identify the region where the Log Service project to be accessed resides (such as China East 1 (Hangzhou) or China North 1 (Qingdao)) and then select the Log Service endpoint that matches with the region to initialize the client. The service endpoint is defined in the same way as the service endpoint of APIs.

Note:

- When selecting an endpoint for the client, make sure that the region where the project to be accessed resides is the same as the region that corresponds to the endpoint. Otherwise, you cannot use SDK to access your specified project.
- The client can only specify the service endpoint when being built, you must use different endpoints to build different clients if you want to access projects in different regions.
- Currently, all the API service endpoints only support HTTP.

- You can also use an intranet endpoint to avoid Internet bandwidth overhead if you are using SDKs in an Alibaba Cloud Elastic Compute Service (ECS) instance. For more information, see Service endpoint.

AccessKey

As described in AccessKey, all requests that interact with Log Service must undergo security verification. An AccessKey is a critical factor in request security verification and is composed of an AccessKey ID and an AccessKey Secret. You must specify two parameters (AccessKey ID and AccessKey Secret), that is, the AccessKey, when building the client. Therefore, log on to the Alibaba Cloud **Access Key Management** page to obtain or create an AccessKey before using SDKs.

Note:

- If you have multiple AccessKeys under your Alibaba Cloud account, make sure that the AccessKey ID and AccessKey Secret specified when building the client are in pair. Otherwise, the AccessKey cannot pass the security verification required by Log Service.
- The specified AccessKey must be enabled. Otherwise, the request is denied by Log Service. You can also log on to the Alibaba Cloud **Access Key Management** page to view the AccessKey status.

Example

To access a project in region China East 1 (Hangzhou) and you have an enabled AccessKey as follows:

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

You can create the client in the following ways:

Java

```
String endpoint = "cn-hangzhou.log.aliyuncs.com";       //The Log Service endpoint of region China East 1
(Hangzhou).
String accessKeyId = "bq2sjzesjmo86kq35behupbq"; //Your AccessKey ID.
String accessKeySecret = "4fdO2fTDDnZPU/L7CHNdemB2Nsk=";//Your AccessKey Secret.

Client client = new Client(endpoint, accessKeyId, acccessKeySecret);

//Use client to operate Log Service project...
```

.NET(C#)

```
String endpoint = "cn-hangzhou.log.aliyuncs.com";       //The Log Service endpoint of region China East 1
(Hangzhou).
String accessKeyId = "bq2sjzesjmo86kq35behupbq"; //Your AccessKey ID.
String accessKeySecret = "4fdO2fTDDnZPU/L7CHNdemB2Nsk=";//Your AccessKey Secret.

SLSClient client = new SLSClient(endpoint, accessKeyId, accessKeySecret);

//Use client to operate Log Service project...
```

PHP

```
$endpoint = 'cn-hangzhou.log.aliyuncs.com'; //The Log Service endpoint of region China East 1 (Hangzhou).
$accessKeyId = 'bq2sjzesjmo86kq35behupbq'; //Your AccessKey ID.
$accessKey = '4fdO2fTDDnZPU/L7CHNdemB2Nsk=';//Your AccessKey Secret.

$client = new Aliyun_Sls_Client($endpoint, $accessKeyId, $accessKey);

//Use client to operate Log Service project...
```

Python

```
# The Log Service endpoint of region China East 1 (Hangzhou).
endpoint = 'cn-hangzhou.log.aliyuncs.com'
# Your AccessKey ID.
accessKeyId = 'bq2sjzesjmo86kq35behupbq'
# Your AccessKey Secret.
accessKey = '4fdO2fTDDnZPU/L7CHNdemB2Nsk='

client = LogClient(endpoint, accessKeyId, accessKey)

#Use client to operate Log Service project...
```

Handle errors

Possible SDK errors are classified as follows:

- Errors returned by Log Service. This type of errors is returned by Log Service and handled by SDKs. For more information about this error type, see the common error codes of the Log Service APIs and the descriptions of each API.
- Network errors that occur when SDKs send requests to Log Service. This type of errors

includes network interruptions and Log Service return timeout.
- Errors that are produced by SDKs and related to platforms and languages, for example, memory overflow.

Currently, SDKs in each language handle errors by throwing exceptions. The specific principles are as follows:

- The first and second types of errors are encapsulated as the LogException class and thrown to users by SDKs.
- The third type of errors is not handled by SDKs, but is thrown to users as the platform- and language-specific Native Exception class.

LogException

The LogException class is defined by SDKs to handle the logical errors of Log Service. It inherits from the basic exception classes of each language and provides the following exception information:

- Error code: Indicates the error type. For errors returned by Log Service, the error code is the same as that returned by APIs. For network errors of SDK requests, the error code is "RequestError". For more information, see the complete API reference of each language.
- Error message: Indicates the message that comes with an error. For errors returned by Log Service, the error message is the same as that returned by APIs. For network errors of SDK requests, the error message is "request is failed". For more information, see the complete API reference of each language.
- Request ID: Indicates the request ID in Log Service that corresponds to the current error. This ID is valid only when Log Service returns an error message. Otherwise, it is an empty string. When a request error occurs, you can provide the request ID to the Log Service team to troubleshoot the problem.

Request failure and retry

When you use SDKs to access Log Service, the request may fail because of temporary network interruptions, transmission delay, and slow processing in Log Service. Currently, this type of errors is directly thrown as exceptions and Log Service does not retry the request. Therefore, you must define the processing logic (retry the request or directly report an error) when using SDKs.

Example

Assume that you want to access the project **big-game** in the region China East 1 (Hangzhou) and retry the request for the specified number of times when a network exception occurs. The code snippets in various languages are as follows:

Java

```
//Other code...

String accessId = "your_access_id"; //TODO: Use your Alibaba Cloud AccessKey ID.
String accessKey = "your_access_key"; //TODO: Use your Alibaba Cloud AccessKey Secret.
String project = "big-game";
String endpoint = "cn-hangzhou.sls.aliyuncs.com";

int max_retries = 3;

/*
* Build a client.
*/
Client client = new Client(accessId, accessKey, endpoint);
ListLogStoresRequest lsRequest = new ListLogStoresRequest(project);

for (int i = 0; i < max_retries; i++)
{
try
{
ListLogStoresResponse res = client.ListLogStores(lsRequest)

//TODO: Process the returned response...
}
catch(LogException ex)
{
if (e.GetErrorCode() == "RequestError")
{
if ( i == max_retries - 1)
{
System.out.println("request is still failed after all retries.");
break;
}
else
System.out.println("request error happens, retry it!");
}
else
{
System.out.println("error code :" + e.GetErrorCode());
System.out.println("error message :" + e.GetErrorMessage());
System.out.println("error requestId :" + e.GetRequestId());
break;
}
}
catch(...)
{
System.out.println("unrecoverable exception when listing logstores.");
break;
}
}

//Other code...
```

.NET(C#)

```
//Other code...

String accessId = "your_access_id"; //TODO: Use your Alibaba Cloud AccessKey ID.
String accessKey = "your_access_key"; //TODO: Use your Alibaba Cloud AccessKey Secret.
String project = "big-game";
String endpoint = "cn-hangzhou.sls.aliyuncs.com";

int max_retries = 3;

//Build a client.
SLSClient client = new SLSClient(endpoint, accessId, accessKey);
ListLogstoresRequest request = new ListLogstoresRequest();
request.Project = project;

for (int i = 0; i < max_retries; i++)
{
try
{
ListLogstoresResponse response = client.ListLogstores(request);

//TODO: Process the returned response...
}
catch(LogException ex)
{
if (e.errorCode == "SLSRequestError")
{
if ( i == max_retries - 1)
{
Console.Writeline( “request is still failed after all retries.” );
break;
}
else
{
Console.Writeline("request error happens, retry it!");
}
}
else
{
Console.Writeline("error code :" + e.errorCode;
Console.Writeline("error message :" + e.Message;
Console.Writeline("error requestId :" + e.RequestId;
break;
}
}
catch(...)
{
Console.Writeline("unrecoverable exception when listing logstores.");
break;
}
}

//Other code...
```

PHP

```
<?php

//Other code...

$endpoint = 'cn-hangzhou.sls.aliyuncs.com';
$accessId = 'your_access_id'; // TODO: Use your Alibaba Cloud AccessKey ID.
$accessKey = 'your_access_key'; //TODO: Use your Alibaba Cloud AccessKey Secret.
$maxRetries = 3;

// Build a client.
$client = new Aliyun_Sls_Client($endpoint, $accessId, $accessKey);

$project = 'big-game';
$request = new Aliyun_Sls_Models_ListLogstoresRequest($project);

for($i = 0; $i < $maxRetries; ++$i)
{
try
{
$response = $client->ListLogstores($request);

//TODO: Process the returned response...
}
catch (Aliyun_Sls_Exception $e)
{
if ($e->getErrorCode()=='RequestError')
{
if ($i+1 == $maxRetries)
{
echo "error code :" . $e->getErrorCode() . PHP_EOL;
echo "error message :" . $e->getErrorMessage() . PHP_EOL;
break;
}
echo 'request error happens, retry it!' . PHP_EOL;
}
else
{
echo "error code :" . $e->getErrorCode() . PHP_EOL;
echo "error message :" . $e->getErrorMessage() . PHP_EOL;
echo "error requestId :" . $e->getRequestId() . PHP_EOL;
break;
}
}
catch (Exception $ex)
{
echo 'unrecoverable exception when listing logstores.' . PHP_EOL;
var_dump($ex);
break;
}
}

//Other code...
```

Python

```
//Other code...

endpoint = 'cn-hangzhou.sls.aliyuncs.com'
accessId = 'your_access_id' # TODO: Use your Alibaba Cloud AccessKey ID.
accessKey = 'your_access_key' # TODO: Use your Alibaba Cloud AccessKey Secret.
maxRetries = 3

# Build a client.
client = Client(endpoint, accessId, accessKey)

project = 'big-game'
lsRequest = ListLogstoresRequest(project)

for i in xrange(maxRetries):
try:
res = client.ListLogstores(lsRequest)
# TODO: Process the returned response...
except LogException as e:
if e.getErrorCode() == "RequestError":
if i+1 == maxRetries:
print "error code :" + e.getErrorCode()
print "error message :" + e.getErrorMessage()
break
else:
print "request error happens, retry it!"
else:
print "error code :" + e.getErrorCode()
print "error message :" + e.getErrorMessage()
print "error requestId :" + e.getRequestId()
break
except Exception as e:
print 'unrecoverable exception when listing logstores.'
break

//Other code...
```

Interface regulations

Though SDKs in different languages are implemented differently, all their interfaces comply with the request-response principle, that is, call the API as follows:

1. Build a request by using request parameters.
2. Call the corresponding interface in the SDK and pass in the request in the preceding step.

3. Encapsulate the results returned by the SDK interface into a response and then return the response to the user.

The following code snippets show how to obtain the names of all Logstores in a project based on the preceding process.

Java

```
// Other code...

String accessId = "your_access_id"; //TODO: Use your Alibaba Cloud AccessKey ID.
String accessKey = "your_access_key"; //TODO: Use your Alibaba Cloud AccessKey Secret.
String project = "your_project"; //TODO: Use your project name.
String endpoint = "region_endpoint";//TODO: Use the endpoint that corresponds to the region where your project
resides.

//Build a client.
Client client = new Client(endpoint, accessId, accessKey);

//Use the request parameter "project" to initialize the ListLogstores request class.
ListLogStoresRequest lsRequest = new ListLogStoresRequest(project);

//Use the request to call the ListLogstores interface. The return parameter is the corresponding response.
ListLogStoresResponse res = client.ListLogStores(lsRequest);

//Access the response to retrieve the request results.
ArrayList<String> names = res.GetLogStores();

// Other code...
```

.NET(C#)

```
// Other code...

String accessId = "your_access_id"; //TODO: Use your Alibaba Cloud AccessKey ID.
String accessKey = "your_access_key"; //TODO: Use your Alibaba Cloud AccessKey Secret.
String project = "your_project"; //TODO: Use your project name.
String endpoint = "region_endpoint";//TODO: Use the endpoint that corresponds to the region where your project
resides.

//Build a client.
SLSClient client = new SLSClient(endpoint, accessId, accessKey);

//Use the request parameter "project" to initialize the ListLogstores request class.
ListLogStoresRequest lsRequest = new ListLogStoresRequest();
lsRequest.Project = project;
```

```
//Use the request to call the ListLogstores interface. The return parameter is the corresponding response.
ListLogStoresResponse res = client.ListLogStores(lsRequest);

//Access the response to retrieve the request results.
List<String> names = res.Logstores;

// Other code...
```

PHP

```
// Other code...

$accessId = "your_access_id"; //TODO: Use your Alibaba Cloud AccessKey ID.
$accessKey = "your_access_key"; //TODO: Use your Alibaba Cloud AccessKey Secret.
$project = "your_project"; //TODO: Use your project name.
$endpoint = "region_endpoint";//TODO: Use the endpoint that corresponds to the region where your project resides.

//Build a client.
$client = new Aliyun_Sls_Client($endpoint, $accessId, $accessKey);

//Use the request parameter "project" to initialize the ListLogstores request class.
$request = new Aliyun_Sls_Models_ListLogstoresRequest($project);

//Use the request to call the ListLogstores interface. The return parameter is the corresponding response.
$response = $client->listLogstores($request);

//Access the response to retrieve the request results.
$names = $response->getLogstores();

// Other code...
```

Python

```
// Other code...

accessId = 'your_access_id'; //TODO: Use your Alibaba Cloud AccessKey ID.
accessKey = 'your_access_key'; //TODO: Use your Alibaba Cloud AccessKey Secret.
project = 'your_project'; //TODO: Use your project name.
endpoint = 'region_endpoint';//TODO: Use the endpoint that corresponds to the region where your project resides.

# Build a client.
client = LogClient(endpoint, accessId, accessKey)

# Use the request parameter "project" to initialize the ListLogstores request class.
lsRequest = ListLogstoresRequest(project)

# Use the request to call the ListLogstores interface. The return parameter is the corresponding response.
res = client.list_logstores(lsRequest)
```

```
# Access the response to retrieve the request results.
names = res.get_logstores();

// Other code...
```

SDKs implement multiple sets of interfaces similar to ListLogstores and define the corresponding request and response classes. In addition to the basic request-response interfaces, SDKs in different languages provide secondary interfaces encapsulated with these basic interfaces, removing the need to build requests and parse the final response on your own. For more information about the secondary interfaces, see the API reference of each SDK.

Java SDK

Download address

Log Service Java SDK allows Java developers to conveniently use Alibaba Cloud Log Service by using the Java programs. You can directly use Maven dependencies to add the SDK or download the package to your local machine. Currently, Log Service Java SDK supports J2SE 6.0 or later versions. Click here to download the latest SDK.

Procedure

Follow these steps to start using the Log Service Java SDK quickly.

Step 1 Create an Alibaba Cloud account

For more information, see Sign up with Alibaba Cloud.

Step 2 Obtain an Alibaba Cloud AccessKey

Before using Log Service Java SDK, you must apply for an Alibaba Cloud AccessKey.

Log on to the Alibaba Cloud **Access Key Management** page. Select an AccessKey for SDK. If you do not have any, create one and make sure the AccessKey is enabled. For how to create an AccessKey, see **Create and enable AccessKey** in Preparation.

The AccessKey is used in the following steps and must be kept confidential. For more information about how to use the AccessKey in SDK, see SDK configuration.

Step 3 Create a Log Service project and a Logstore

Before using Log Service Java SDK, you must create a Log Service project and a Logstore in the console.

For how to create a project and a Logstore, see **Create a project** in Manage a project and **Create a Logstore** in Manage a Logstore.

Note:

- Make sure that you use the same Alibaba Cloud account to obtain the Alibaba Cloud AccessKey and create the Log Service project and Logstore.
- For more information about the concepts of Log Service such as project and Logstore, see Core concept.
- A project name must be globally unique in Log Service, and a Logstore name must be unique in the same project.
- After a project is created, you cannot modify the region or migrate the project across regions.

Step 4 Install the Java development environment

Currently, Log Service Java SDK supports the Java runtime environment of J2SE 6.0 or later versions. You can download the installation package at the Java official website and follow the instructions to install the Java development environment.

Step 5 Install Log Service Java SDK

You must install the Log Service Java SDK after installing the Java development environment. Currently, you can install the Log Service Java SDK in the following two ways:

We recommend that you use Apache Maven to obtain the latest SDK version. You can add the following configurations to your Maven project.

```
<dependency>
<groupId>com.google.protobuf</groupId>
<artifactId>protobuf-java</artifactId>
<version>2.5.0</version>
</dependency>
<dependency>
<groupId>com.aliyun.openservices</groupId>
<artifactId>aliyun-log</artifactId>
<version>0.6.7</version>
<exclusions>
<exclusion>
<groupId>com.google.protobuf</groupId>
<artifactId>protobuf-java</artifactId>
```

```
</exclusion>
</exclusions>
</dependency>
```

You can also download the Java SDK package and then directly reference the local software package in your Java project.

i. Click here to clone the Java SDK package. (Version updates are provided periodically. Use Maven to obtain the latest version.)
ii. Decompress the downloaded package to the specified directory. The Java SDK is a software development kit and does not require additional installation.
iii. Add all .jar packages (including third-party dependent packages) in the SDK package to your Java project. (For more information, see the corresponding IDE document.)

Step 6 Start a new Java project

You can start using the Java SDK. To interact with Log Service and obtain the relevant output, run the following sample code in a text editor or Java IDE. For more information about the instructions when using Java SDK, see **Instructions** in this document.

```
package sdksample;

import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import java.util.Date;
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.*;
import com.aliyun.openservices.log.exception.*;
import com.aliyun.openservices.log.request.*;
import com.aliyun.openservices.log.response.*;
import com.aliyun.openservices.log.common.LogGroupData;
import com.aliyun.openservices.log.common.LogItem;
import com.aliyun.openservices.log.common.Logs.Log;
import com.aliyun.openservices.log.common.Logs.Log.Content;
import com.aliyun.openservices.log.common.Logs.LogGroup;
import com.aliyun.openservices.log.common.Consts.CursorMode;
import com.facebook.presto.jdbc.internal.guava.collect.ImmutableList;

public class sdksample {

public static void main(String args[]) throws LogException, InterruptedException {
String endpoint = "<log_service_endpoint>"; // Select the endpoint that matches the region of the project created
in the preceding step.
String accessKeyId = "<your_access_key_id>"; // Use your Alibaba Cloud AccessKey ID.
String accessKeySecret = "<your_access_key_secret>"; // Use your Alibaba Cloud AccessKey Secret.
String project = "<project_name>"; // The name of the project created in the preceding step.
String logstore = "<logstore_name>"; // The name of the Logstore created in the preceding step.
// Build a client.
Client client = new Client(endpoint, accessKeyId, accessKeySecret);
```

```
// List the names of all the Logstores in the current project.
int offset = 0;
int size = 100;
String logStoreSubName = "";
ListLogStoresRequest req1 = new ListLogStoresRequest(project, offset, size, logStoreSubName);
ArrayList<String> logStores = client.ListLogStores(req1).GetLogStores();
System.out.println("ListLogs:" + logStores.toString() + "\n");
// Write logs.
String topic = "";
String source = "";
// Send 10 packets consecutively, with each packet containing 10 logs.
for (int i = 0; i < 10; i++) {
Vector<LogItem> logGroup = new Vector<LogItem>();
for (int j = 0; j < 10; j++) {
LogItem logItem = new LogItem((int) (new Date().getTime() / 1000));
logItem.PushBack("index"+String.valueOf(j), String.valueOf(i * 10 + j));
logGroup.add(logItem);
}
PutLogsRequest req2 = new PutLogsRequest(project, logstore, topic, source, logGroup);
client.PutLogs(req2);
/*
* You can specify the shard to which data is sent by setting the shard HashKey. Data is written to the shard whose range includes the HashKey. For more information about the API, see the following interface: public PutLogsResponse
* PutLogs( String project, String logStore, String topic,
* List<LogItem> logItems, String source, String shardHash // Write data to the shard based on the hashkey, which may be MD5(ip) or MD5(id).) throws
* LogException;
*/
}
// Read the data written to shard 0 in the last one minute.
int shard_id = 0;
long curTimeInSec = System.currentTimeMillis() / 1000;
GetCursorResponse cursorRes = client.GetCursor(project, logstore, shard_id, curTimeInSec - 60);
String beginCursor = cursorRes.GetCursor();
cursorRes = client.GetCursor(project, logstore, shard_id, CursorMode.END);
String endCursor = cursorRes.GetCursor();
String curCursor = beginCursor;
while (curCursor.equals(endCursor) == false) {
int loggroup_count = 2; // Read two log groups at a time.
BatchGetLogResponse logDataRes = client.BatchGetLog(project, logstore, shard_id, loggroup_count, curCursor, endCursor);
// Read the log group list.
List<LogGroupData> logGroups = logDataRes.GetLogGroups();
for(LogGroupData logGroup: logGroups){
FastLogGroup flg = logGroup.GetFastLogGroup();
System.out.println(String.format("\tcategory\t:\t%s\n\tsource\t:\t%s\n\ttopic\t:\t%s\n\tmachineUUID\t:\t%s", flg.getCategory(), flg.getSource(), flg.getTopic(), flg.getMachineUUID()));
System.out.println("Tags");
for (int tagIdx = 0; tagIdx < flg.getLogTagsCount(); ++tagIdx) {
FastLogTag logtag = flg.getLogTags(tagIdx);
System.out.println(String.format("\t%s\t:\t%s", logtag.getKey(), logtag.getValue()));
}
for (int lIdx = 0; lIdx < flg.getLogsCount(); ++lIdx) {
FastLog log = flg.getLogs(lIdx);
System.out.println("--------\nLog: " + lIdx + ", time: " + log.getTime() + ", GetContentCount: " +
log.getContentsCount());
for (int cIdx = 0; cIdx < log.getContentsCount(); ++cIdx) {
FastLogContent content = log.getContents(cIdx);
System.out.println(content.getKey() + "\t:\t" + content.getValue());
}
}
}
String next_cursor = logDataRes.GetNextCursor();
System.out.println("The Next cursor:" + next_cursor);
curCursor = next_cursor;
}
// Note: You can call the following interface only after the index function is enabled.
// Wait one minute until logs can be queried.
try {
Thread.sleep(60 * 1000);
} catch (InterruptedException e) {
e.printStackTrace();
}
// Query log distribution.
String query = "<The query keyword. To query all the contents, use an empty string here.>";
int from = (int) (new Date().getTime() / 1000 - 300);
int to = (int) (new Date().getTime() / 1000);
GetHistogramsResponse res3 = null;
while (true) {
GetHistogramsRequest req3 = new GetHistogramsRequest(project, logstore, topic, query, from, to);
res3 = client.GetHistograms(req3);
if (res3 != null && res3.IsCompleted()) // If IsCompleted() returns "true", the query results are accurate. If "false" is
returned, query the results again.
{
break;
}
Thread.sleep(200);
}
System.out.println("Total count of logs is " + res3.GetTotalCount());
for (Histogram ht : res3.GetHistograms()) {
System.out.printf("from %d, to %d, count %d.\n", ht.GetFrom(), ht.GetTo(), ht.GetCount());
}
// Query log data.
long total_log_lines = res3.GetTotalCount();
int log_offset = 0;
int log_line = 10;
while (log_offset <= total_log_lines) {
GetLogsResponse res4 = null;
// Read 10 lines of logs at a time for each log offset. If failed, retry at most three times.
for (int retry_time = 0; retry_time < 3; retry_time++) {
GetLogsRequest req4 = new GetLogsRequest(project, logstore, from, to, topic, query, log_offset,
log_line, false);
res4 = client.GetLogs(req4);
if (res4 != null && res4.IsCompleted()) {
break;
}
Thread.sleep(200);
}
System.out.println("Read log count:" + String.valueOf(res4.GetCount()));
log_offset += log_line;
}
```

```
// Enable the analysis function. You can use the SQL function only after enabling the analysis function. You can
enable the analysis function in the console or by using SDKs.
IndexKeys indexKeys = new IndexKeys();
ImmutableList.Builder<String> tokens = ImmutableList.builder();
tokens.add(",");
tokens.add(".");
tokens.add("#");
IndexKey keyContent = new IndexKey(tokens.build(),false,"text");
indexKeys.AddKey("index0",keyContent);
keyContent = new IndexKey(ImmutableList.<String>builder().build(),false,"long");
indexKeys.AddKey("index1",keyContent);
keyContent = new IndexKey(ImmutableList.<String>builder().build(),false,"double");
indexKeys.AddKey("index2",keyContent);

IndexLine indexLine = new IndexLine(ImmutableList.<String>builder().build(),false);
Index index = new Index(7,indexKeys,indexLine);
CreateIndexRequest createIndexRequest = new CreateIndexRequest(project,logstore,index);
client.CreateIndex(createIndexRequest);


// Use the analysis function.
GetLogsRequest req4 = new GetLogsRequest(project, logstore, from, to, "", " index0:value | select avg(index1) as
v1,sum(index2) as v2, index0 group by index0");
GetLogsResponse res4 = client.GetLogs(req4);
if(res4 != null && res4.IsCompleted()){
for (QueriedLog log : res4.GetLogs()){
LogItem item = log.GetLogItem();
for(LogContent content : item.GetLogContents()){
System.out.print(content.GetKey()+":"+content.GetValue());
}
System.out.println();
}
}
}
}
```

Instructions

To improve the I/O efficiency of your system, try not to directly use SDKs to write data to Log Service. For more information about the standard way to write data, see Producer Library.

To consume data in Log Service, try not to directly use SDKs to pull data interfaces. An advanced consumer library is provided, which shields the implementation details of Log Service and provides the advanced functions such as load balancing and consumption in order.

.NET SDK

Log Service .NET SDK allows developers of Windows platform to conveniently use Alibaba Cloud Log Service by using the .NET platform. Currently, the SDK supports the .NET Framework 3.5, 4.0, and 4.5. SDK files vary with different .NET Framework versions, but the interfaces and functions are the same.

Download address

SDK GitHub address: https://github.com/aliyun/aliyun-log-csharp-sdk

Procedure

Follow these steps to start using the Log Service .NET SDK quickly.

Step 1 Create an Alibaba Cloud account

For more information, see Sign up with Alibaba Cloud.

Step 2 Obtain an Alibaba Cloud AccessKey

Before using Log Service .NET SDK, you must apply for an Alibaba Cloud AccessKey.

Log on to the Alibaba Cloud **Access Key Management** page. Select an AccessKey for SDK. If you do not have any, create one and make sure the AccessKey is enabled. For how to create an AccessKey, see **Create and enable AccessKey** in Preparation.

The AccessKey is used in the following steps and must be kept confidential. For more information about how to use the AccessKey in SDK, see SDK configuration.

Step 3 Create a Log Service project and a Logstore

Before using Log Service .NET SDK, you must create a Log Service project and a Logstore in the console.

For how to create a project and a Logstore, see **Create a project** in Manage a project and **Create a Logstore** in Manage a Logstore.

Note:

- Make sure that you use the same Alibaba Cloud account to obtain the Alibaba Cloud AccessKey and create the Log Service project and Logstore.
- For more information about the concepts of Log Service such as project and Logstore,

see Core concept.
- A project name must be globally unique in Log Service, and a Logstore name must be unique in the same project.
- After a project is created, you cannot modify the region or migrate the project across regions.

Step 4 Install the .NET development environment

Currently, Log Service SDK supports the .NET 3.5 and .NET 4.0/4.5 running environments. To support the Log Service SDK development, we recommend that you install:

- Microsoft .NET Framework 3.5/4.0/4.5 (the specific version depends on the target environment required by your program.)
- Visual Studio 2010 and later versions

Step 5 Download and install Log Service .NET SDK

Install the Log Service .NET SDK after you build the .NET development environment. The steps are as follows:

- Download from GitHub.
- Historical version download: Click here to download the Log Service .NET SDK package of the latest version.

Step 6 Start a new Log Service .NET project

After installing the .NET development environment and the Log Service .NET SDK, you can create a Log Service .NET project. For more information, see the LOGSDKSample project of SLSSDK40 solution in GitHub.

PHP SDK

Log Service PHP SDK allows PHP developers to use Alibaba Cloud Log Service conveniently.

Download address

SDK GitHub address: https://github.com/aliyun/aliyun-log-php-sdk

Procedure

Follow these steps to start using the Log Service PHP SDK quickly.

Step 1 Create an Alibaba Cloud account

For more information, see Sign up with Alibaba Cloud.

Step 2 Obtain an Alibaba Cloud AccessKey

Before using Log Service PHP SDK, you must apply for an Alibaba Cloud AccessKey.

Log on to the Alibaba Cloud **Access Key Management** page. Select an AccessKey for SDK. If you do not have any, create one and make sure the AccessKey is enabled. For how to create an AccessKey, see **Create and enable AccessKey** in Preparation.

The AccessKey is used in the following steps and must be kept confidential. For more information about how to use the AccessKey in SDK, see SDK configuration.

Step 3 Create a Log Service project and a Logstore

Before using Log Service PHP SDK, you must create a Log Service project and a Logstore in the console.

For how to create a project and a Logstore, see **Create a project** in Manage a project and **Create a Logstore** in Manage a Logstore.

Note:

- Make sure that you use the same Alibaba Cloud account to obtain the Alibaba Cloud AccessKey and create the Log Service project and Logstore.
- For more information about the concepts of Log Service such as project and Logstore, see Core concept.
- A project name must be globally unique in Log Service, and a Logstore name must be unique in the same project.
- After a project is created, you cannot modify the region or migrate the project across regions.

Step 4 Install the PHP development environment

The PHP SDK supports PHP 5.2.1 and later versions. You can install any of these versions locally and build the corresponding PHP development environment.

Step 5 Download and install PHP SDK

You must install the PHP SDK after building the PHP development environment. Follow these steps:

1. Download the latest PHP SDK package from GitHub.
2. Decompress the downloaded package to the specified directory. The PHP SDK is a software development kit and does not require additional installation. In addition to the SDK codes, the SDK has a set of third-party dependent packages and an autoloader class for simplified use. Follow these steps to use the SDK directly in your PHP project.

Step 6 Start a new PHP project

You can start using the PHP SDK. To interact with Log Service and obtain the relevant output, run the following sample code in a text editor or PHP IDE.

```
<?php

/* Use the autoloader class to automatically load all required PHP modules. Specify the proper path of the file
containing the autoloader class.*/
require_once realpath(dirname(__FILE__) . '/../Log_Autoload.php');

$endpoint = 'cn-hangzhou.sls.aliyuncs.com'; // Select the endpoint that matches the region of the project created
in the preceding step.
$accessKeyId = 'your_access_key_id'; // Use your Alibaba Cloud AccessKey ID.
$accessKey = 'your_access_key'; // Use your Alibaba Cloud AccessKey Secret.
$project = 'your_project'; // The name of the project created in the preceding step.
$logstore = 'your_logstore'; // The name of the Logstore created in the preceding step.

$client = new Aliyun_Log_Client($endpoint, $accessKeyId, $accessKey);

#List the names of all the Logstores in the current project.
$req1 = new Aliyun_Log_Models_ListLogstoresRequest($project);
$res1 = $client->listLogstores($req1);
var_dump($res1);


#Create a Logstore.
$req2 = new Aliyun_Log_Models_CreateLogstoreRequest($project,$logstore,3,2);
$res2 = $client -> createLogstore($req2);

#Wait until the Logstore takes effect.
sleep(60);


#Write logs.
$topic = "";
$source = "";
$logitems = array();
for ($i = 0; $i < 5; $i++)
{
$contents = array('index1'=> strval($i));
$logItem = new Aliyun_Log_Models_LogItem();
$logItem->setTime(time());
$logItem->setContents($contents);
array_push($logitems, $logItem);
}
$req2 = new Aliyun_Log_Models_PutLogsRequest($project, $logstore, $topic, $source, $logitems);

$res2 = $client->putLogs($req2);
var_dump($res2);


#Drag data immediately.
#Traverse shard IDs.
$listShardRequest = new Aliyun_Log_Models_ListShardsRequest($project,$logstore);

$listShardResponse = $client -> listShards($listShardRequest);

foreach($listShardResponse-> getShardIds() as $shardId)

{

#Obtain the cursor corresponding to each shard ID.

$getCursorRequest = new Aliyun_Log_Models_GetCursorRequest($project,$logstore,$shardId,null, time() - 60);
$response = $client -> getCursor($getCursorRequest);
$cursor = $response-> getCursor();
$count = 100;
while(true)
{
#Read data starting from the cursor.
$batchGetDataRequest = new
Aliyun_Log_Models_BatchGetLogsRequest($project,$logstore,$shardId,$count,$cursor);
var_dump($batchGetDataRequest);
$response = $client -> batchGetLogs($batchGetDataRequest);
if($cursor == $response -> getNextCursor())
{
break;
}
$logGroupList = $response -> getLogGroupList();
foreach($logGroupList as $logGroup)
{
print ($logGroup->getCategory());

foreach($logGroup -> getLogsArray() as $log)
{
foreach($log -> getContentsArray() as $content)
{
print($content-> getKey().":".$content->getValue()."\t");
}
print("\n");
}
}
$cursor = $response -> getNextCursor();
}
}
```

```
#Wait one minute until logs can be queried.
sleep(60);

#Query log distribution (Note: Make sure that indexes are created before you query logs. The PHP SDK does not
provide this interface, so you must create them in the console.)
$topic = "";
$query='';
$from = time()-3600;
$to = time();

$res3 = NULL;
while (is_null($res3) || (! $res3->isCompleted()))
{
$req3 = new Aliyun_Log_Models_GetHistogramsRequest($project, $logstore, $from, $to, $topic, $query);
$res3 = $client->getHistograms($req3);
}

var_dump($res3);


#Query log data.
$res4 = NULL;

while (is_null($res4) || (! $res4->isCompleted()))
{
$req4 = new Aliyun_Log_Models_GetLogsRequest($project, $logstore, $from, $to, $topic, $query, 5, 0, False);
$res4 = $client->getLogs($req4);
}
var_dump($res4);
```

Python SDK

Download address

Click here to download Python SDK.

SDK GitHub address: https://github.com/aliyun/aliyun-log-python-sdk

Procedure

Follow these steps to start using the Log Service Python SDK quickly.

Step 1 Create an Alibaba Cloud account

For more information, see Sign up with Alibaba Cloud.

Step 2 Obtain an Alibaba Cloud AccessKey

Before using Log Service Python SDK, you must apply for an Alibaba Cloud AccessKey.

Log on to the Alibaba Cloud **Access Key Management** page.

Select an AccessKey for SDK. If you do not have any, create one and make sure the AccessKey is enabled.

The AccessKey is used in the following steps and must be kept confidential. For more information about how to use the AccessKey in SDK, see SDK configuration.

Step 3 Create a Log Service project and a Logstore

Before using Log Service Python SDK, you must create a Log Service project and a Logstore in the console.

Log on to the Log Service console.

Click **Create Project** in the upper-right corner.

Enter the **Project Name** and select the **Region**. Click **Confirm**.

To create a Logstore, click **Create** in the displayed dialog box after creating a project, or click the project name on the **Project List** page and then click **Create** in the upper-right corner.

Complete the configurations and then click **Confirm**.

Enter the **Logstore Name** and **Data Retention Time**. Select the **Number of Shards** as per your needs. In this example, you must configure four shards.

Note:

- Make sure that you use the same Alibaba Cloud account to obtain the Alibaba Cloud AccessKey and create the Log Service project and Logstore.
- For more information about the concepts of Log Service such as project and Logstore, see Core concept.
- A project name must be globally unique in Log Service, and a Logstore name must be

unique in the same project.
- After a project is created, you cannot modify the region or migrate the project across regions.

Step 4 Install Python environment

The Python SDK is a pure Python library and supports all operating systems that Python can run, including Linux, Mac OS X, and Windows.

Install Python as follows:

Download and install the latest Python installation package.

Note:

- Currently, Python SDK supports the Python 2.6/2.7 and Python 3.3/3.4/3.5/3.6 environments. You can run the python -V command to query the current running version of Python.
- Python does not officially support Python 2.6 or Python 3.3. We recommend that you use Python 2.7, Python 3.4, and later versions.

Download and install the Python package management tool pip.

After pip is installed, run pip -V to check whether or not the installation is successful and query the current pip version.

Step 5 Install Python SDK

Run the following command as an administrator in Shell to install Python SDK.

```
pip install -U aliyun-log-python-sdk
```

Step 6 Start a Python program

You can start using the Python SDK. To interact with Log Service and obtain the relevant output, run the following sample code in a text editor or Python IDE.

For more information, see Github/reamdthedocs.

```
# encoding: utf-8

import time
```

```
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
from aliyun.log.getlogsrequest import GetLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest


def main():
endpoint = '' # Select the endpoint that matches the region of the project created in the preceding step.
accessKeyId = '' # Use your Alibaba Cloud AccessKey ID.
accessKey = '' # Use your Alibaba Cloud AccessKey Secret.
project = '' # The name of the project created in the preceding step.
logstore = '' # The name of the Logstore created in the preceding step.

# Note: Configure four shards for the created Logstore for later testing.

# Build a client.
client = LogClient(endpoint, accessKeyId, accessKey)

# List all Logstores.
req1 = ListLogstoresRequest(project)
res1 = client.list_logstores(req1)
res1.log_print()
topic = ""
source = ""

# Send 10 data packets, each of which containing 10 logs.
for i in range(10):
logitemList = [] # LogItem list
for j in range(10):
contents = [('index', str(i * 10 + j))]
logItem = LogItem()
logItem.set_time(int(time.time()))
logItem.set_contents(contents)
logitemList.append(logItem)
req2 = PutLogsRequest(project, logstore, topic, source, logitemList)
res2 = client.put_logs(req2)
res2.log_print()

# List all shards and read the data written in the last minute.
listShardRes = client.list_shards(project, logstore)
for shard in listShardRes.get_shards_info():
shard_id = shard["shardID"]
start_time = int(time.time() - 60)
end_time = start_time + 60
res = client.get_cursor(project, logstore, shard_id, start_time)
res.log_print()
start_cursor = res.get_cursor()
res = client.get_cursor(project, logstore, shard_id, end_time)
end_cursor = res.get_cursor()

while True:
loggroup_count = 100 # Read 100 packets each time.
res = client.pull_logs(project, logstore, shard_id, start_cursor, loggroup_count, end_cursor)
res.log_print()
```

```
next_cursor = res.get_next_cursor()
if next_cursor == start_cursor:
break
start_cursor = next_cursor

# Note: You can use the following interfaces to query data only when the index function is enabled.
time.sleep(60)
topic = ""
query = "index"
From = int(time.time()) - 600
To = int(time.time())
res3 = None

# Query the number of logs that match the query conditions in the last 10 minutes. Retry if not all results are correct.
while (res3 is None) or (not res3.is_completed()):
req3 = GetHistogramsRequest(project, logstore, From, To, topic, query)
res3 = client.get_histograms(req3)
res3.log_print()

# Obtain the number of logs that match the query conditions.
total_log_count = res3.get_total_count()
log_line = 10

# Read 10 logs each time until all log data is queried. Retry three times if not all query results are correct during each query.
for offset in range(0, total_log_count, log_line):
res4 = None
for retry_time in range(0, 3):
req4 = GetLogsRequest(project, logstore, From, To, topic, query, log_line, offset, False)
res4 = client.get_logs(req4)
if res4 is not None and res4.is_completed():
break
time.sleep(1)
if res4 is not None:
res4.log_print()
listShardRes = client.list_shards(project, logstore)
shard = listShardRes.get_shards_info()[0]

# Split shards.
if shard["status"] == "readwrite":
shard_id = shard["shardID"]
inclusiveBeginKey = shard["inclusiveBeginKey"]
midKey = inclusiveBeginKey[:-1] + str((int(inclusiveBeginKey[-1:])) + 1)
client.split_shard(project, logstore, shard_id, midKey)

# Merge shards.
shard = listShardRes.get_shards_info()[1]
if shard["status"] == "readwrite":
shard_id = shard["shardID"]
client.merge_shard(project, logstore, shard_id)

# Delete shards.
shard = listShardRes.get_shards_info()[-1]
if shard["status"] == "readonly":
shard_id = shard["shardID"]
```

```
client.delete_shard(project, logstore, shard_id)


if __name__ == '__main__':
main()
```

Android SDK

Alibaba Cloud Log Service Android SDK is mainly used to solve the issues of collecting user data on the Android platform and currently provides the log writing function.

GitHub address: https://github.com/aliyun/aliyun-log-android-sdk

C SDK

Alibaba Cloud Log Service C SDK is mainly used to solve the log access problems in various platforms, for example, to be compatible with MIPS chip and OpenWrt system.

The C SDK uses libCurl as a network library. The apr/apr-util library solves the issues of memory and different platforms. You can use the C SDK after the compilation based on the source codes.

In addition, the C Producer Library and C Producer Lite Library provide you with a one-stop log collection solution that is simple, high-availability, low resource-consuming, and applicable to different platforms.

For the GitHub project addresses, see:

- C Producer Library (recommended for servers)
- C Producer Lite Library (recommended for IOT and smart devices)
- Native APIs of C SDK (recommended for secondary development)

Go

Alibaba Cloud Log Service Go SDK supports:

- Writing and consuming data in batches.
- Querying and indexing data by using keywords.
- Managing Logtail configurations and machine groups.

GitHub address: https://github.com/aliyun/aliyun-log-go-sdk

For more information about the usage instructions and sample codes, see the README.

iOS SDK

Alibaba Cloud Log Service SDKs are implemented based on APIs and currently provide the log writing function.

GitHub address:

- Swift: https://github.com/aliyun/aliyun-log-ios-sdk
- Object-C: https://github.com/lujiajing1126/AliyunLogObjc (Third-party SDKs that are verified in the testing and production environments. You can raise an issue to the author and get a quick response.)

Swift

```
 /*
Use the endpoint, AccessKey ID, and AccessKey Secret to build the Log Service client.
@endPoint: The service endpoint.
*/
let myClient = try! LOGClient(endPoint: "",
accessKeyID: "",
accessKeySecret: "",
projectName:"")

/* Create a log group. */
let logGroup = try! LogGroup(topic: "mTopic",source: "mSource")

/* Store a log. */
let log1 = Log()
try! log1.PutContent("K11", value: "V11")
try! log1.PutContent("K12", value: "V12")
try! log1.PutContent("K13", value: "V13")
logGroup.PutLog(log1)

/* Store a log. */
let log2 = Log()
try! log2.PutContent("K21", value: "V21")
try! log2.PutContent("K22", value: "V22")
try! log2.PutContent("K23", value: "V23")
```

```
logGroup.PutLog(log2)

/* Send the log. */
myClient.PostLog(logGroup,logStoreName: ""){ response, error in

// handle response however you want
if error?.domain == NSURLErrorDomain && error?.code == NSURLErrorTimedOut {
print("timed out") // note, `response` is likely `nil` if it timed out
}
}
```

Objective-C

See GitHub.

Command Line Interface - CLI

Log Service Command Line Interface (CLI) supports almost all operations as web. It also supports integrity check, automatic paging, multiple accounts, and cross-domain replication.

Download link

Click here to download CLI.

For more information, see GitHub document.

Major features

- Supports most of the Log Service REST APIs.
- Supports multiple accounts to facilitate testing and cross-domain operations.
- Supports integrity check and automatic paging for log query.
- Supports storing and configuring AccessKey in multiple ways to apply to various cases.
- Supports inputting complex parameters by using command lines or files and verifying the contents and formats of the parameters.
- Supports using JMES filter to further process the results for easy selection of specific information.
- Supports multiple platforms (Windows, Linux, and Mac).
- Based on Python (py 2.6+ and py 3.3+).
- Supports installing pip.