

Alibaba Cloud Threat Detection

API Reference

Issue: 20190917

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use

or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 Note: You can use Ctrl + A to select all files.
>	Multi-level menu cascade.	Settings > Network > Set network type
Bold	It is used for buttons, menus, page names, and other UI elements.	Click OK.
<code>Courier</code> font	It is used for commands.	Run the <code>cd / d C :/ windows</code> command to enter the Windows system folder.
<i>Italics</i>	It is used for parameters and variables.	<code>bae log list --instanceid Instance_ID</code>
[] or [a b]	It indicates that it is an optional value, and only one item can be selected.	<code>ipconfig [-all -t]</code>

Style	Description	Example
<code>{}</code> or <code>{a b}</code>	It indicates that it is a required value, and only one item can be selected.	<code>switch {stand slave}</code>

Contents

Legal disclaimer.....	I
Generic conventions.....	I
1 Log Analysis API.....	1
1.1 Overview.....	1
1.2 Log analysis Logstore and Project.....	2
1.3 Service endpoint.....	2
1.4 AccessKey.....	6
1.5 Public request header.....	6
1.6 Public response header.....	10
1.7 Request signature.....	11
1.8 Common error codes.....	17
1.9 Project interfaces.....	20
1.9.1 CreateProject.....	20
1.9.2 DeleteProject.....	22
1.9.3 UpdateProject.....	23
1.9.4 GetProject.....	25
1.9.5 ListProject.....	27
1.9.6 GetProjectLogs.....	30
1.10 Logstore APIs.....	34
1.10.1 CreateLogstore.....	34
1.10.2 DeleteLogstore.....	36
1.10.3 UpdateLogstore.....	38
1.10.4 GetLogstore	41
1.10.5 ListLogstore.....	43
1.10.6 ListShards.....	45
1.10.7 SplitShard.....	47
1.10.8 MergeShards.....	50
1.10.9 GetCursor.....	53
1.10.10 PullLogs.....	56
1.10.11 PutLogs.....	58
1.10.12 GetShipperStatus.....	62
1.10.13 Retryshippertask.....	67
1.10.14 GetLogs.....	69
1.10.15 GetHistograms.....	75
1.10.16 CreateIndex.....	81
1.10.17 DeleteIndex.....	85
1.10.18 GetIndex.....	87
1.10.19 UpdateIndex.....	91
1.11 Logtail machine group related interfaces.....	96
1.11.1 Createmachinegroup.....	96

1.11.2 DeleteMachineGroup.....	99
1.11.3 UpdateMachineGroup.....	100
1.11.4 ListMachineGroup.....	104
1.11.5 GetMachineGroup.....	106
1.11.6 ApplyConfigToMachineGroup.....	109
1.11.7 RemoveConfigFromMachineGroup.....	111
1.11.8 ListMachines.....	113
1.11.9 GetAppliedConfigs.....	116
1.12 Logtail configuration related interfaces.....	118
1.12.1 CreateConfig.....	118
1.12.2 ListConfig.....	125
1.12.3 GetAppliedMachineGroups.....	127
1.12.4 GetConfig.....	129
1.12.5 DeleteConfig.....	134
1.12.6 UpdateConfig.....	135
1.13 Consumer group interfaces.....	141
1.13.1 CreateConsumerGroup.....	141
1.13.2 DeleteConsumerGroup.....	144
1.13.3 GetCheckPoint.....	146
1.13.4 HeartBeat.....	149
1.13.5 ListConsumerGroup.....	151
1.13.6 UpdateCheckPoint.....	153
1.13.7 UpdateConsumerGroup.....	156
1.14 External Storage.....	159
1.14.1 External Store related APIs.....	159
1.14.2 External Store overview.....	166
1.15 RAM/STS.....	166
1.15.1 Overview.....	166
1.15.2 Resource list.....	169
1.15.3 Action list.....	171
1.15.4 Authentication rules.....	174
1.16 STS access mode.....	177
1.16.1 Overview.....	177
1.17 Common resources.....	179
1.17.1 Data model.....	179
1.17.2 Data encoding method.....	183
1.17.3 Logstore.....	184
1.17.4 Shard.....	185
1.17.5 Logtail configuration.....	186
1.17.6 Machine group.....	196

1 Log Analysis API

1.1 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. Currently, the following APIs are available:

Objects	Method
Log	Basic concepts of logs and log groups
Project	List , Create , Delete , Get
Project	#unique_10 . Count all the logs in a project.
Config (Configuration)	List , Create , Delete , Get , Update
	GetAppliedMachineGroups (apply/remove a configuration)
MachineGroup (Machine Group)	List , Create , Delete , Get , Update
	Apply/Remove
	GetAppliedConfigs (query the list of applied configurations)
LogStore (Logstore)	List , Create , Delete , Get , Update
	GetLogs (query logs) and GetHistograms (query log distribution)
Index	Create , Update , Delete#unique_38
Shard (Partition)	List , Split , Merge
	PostLogStoreLogs (write a log)
	GetCursor (locate the log location)
	PullLogs (consume a log)

Objects	Method
Shipper	GetShipperStatus (query the status of a LogShipper task)
	RetryShipperTask (retry a failed LogShipper task)
ConsumerGroup	Create , Update , Delete , List
	HeartBeat , GetCheckpoint , UpdateCheckpoint

You can use the APIs to:

- Collect logs based on [#unique_11](#) and [#unique_18](#).
- Create [#unique_27](#). Then, write and read logs to/from the Logstore.
- Set [access control rules](#) for different users.



Note:

- Currently, APIs provide the [Rest](#) style.
- To use APIs, you must know [#unique_56](#).
- Security verification is required for all API requests. For more information about the signature and process of API requests, see [#unique_57](#).
- 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 HTTP header, see [#unique_58](#), This HTTP header must participate in the signature. For more information, see [#unique_57](#).

1.2 Log analysis Logstore and Project

All logs of the Security Center are stored in `sas - log` dedicated Logstore. You can locate dedicated Logstore in project `sas - log - account ID` that stores logs.

1.3 Service endpoint

Service endpoints for the public network

The Log Service endpoint is a URL used to access a project and logs in the project. It 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 following table lists the public network service endpoints for each region.

Region	Root service endpoint
China (Hangzhou)	cn-hangzhou.log.aliyuncs.com
China (Shanghai)	cn-shanghai.log.aliyuncs.com
China (Qingdao)	cn-qingdao.log.aliyuncs.com
China (Beijing)	cn-beijing.log.aliyuncs.com
China (Zhangjiakou)	cn-zhangjiakou.log.aliyuncs.com
China (Hohhot)	cn-huhehaote.log.aliyuncs.com
China (Shenzhen)	cn-shenzhen.log.aliyuncs.com
China (Chengdu)	cn-chengdu.log.aliyuncs.com
Hong Kong	cn-hongkong.log.aliyuncs.com
Japan (Tokyo)	ap-northeast-1.log.aliyuncs.com
Singapore	ap-southeast-1.log.aliyuncs.com
Australia (Sydney)	ap-southeast-2.log.aliyuncs.com
Malaysia (Kuala Lumpur)	ap-southeast-3.log.aliyuncs.com
Indonesia (Jakarta)	ap-southeast-5.log.aliyuncs.com
UAE (Dubai)	me-east-1.log.aliyuncs.com
US (Silicon Valley)	us-west-1.log.aliyuncs.com
Germany (Frankfurt)	eu-central-1.log.aliyuncs.com
US (Virginia)	us-east-1.log.aliyuncs.com
India (Mumbai)	ap-south-1.log.aliyuncs.com
UK (London)	eu-west-1.log.aliyuncs.com

When accessing a specific project, you need to construct the final service endpoint based on the project name and region. The format is as follows:

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

For example, if the project name is big-game and the region is China (Hangzhou), the service endpoint 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 change the region or migrate the project across regions. You must select a root service endpoint that matches the region to compose the service endpoint for this project. The service endpoint is used for API operation requests.

Service endpoints for the classic network or VPC

You can also use private network service endpoints to call Log Service API operations on an Alibaba Cloud Elastic Compute Service (ECS) instance, including the Virtual Private Cloud (VPC) environment. Access to Log Service through private network service endpoints does not consume the Internet traffic of ECS instances, and therefore saves the Internet bandwidth of ECS instances. The following table lists the private network service endpoints for each region.

Region	Root service endpoint
China (Hangzhou)	cn-hangzhou-intranet.log.aliyuncs.com
China (Shanghai)	cn-shanghai-intranet.log.aliyuncs.com
China (Qingdao)	cn-qingdao-intranet.log.aliyuncs.com
China (Beijing)	cn-beijing-intranet.log.aliyuncs.com
China (Shenzhen)	cn-shenzhen-intranet.log.aliyuncs.com
China (Zhangjiakou)	cn-zhangjiakou-intranet.log.aliyuncs.com
China (Hohhot)	cn-huhehaote-intranet.log.aliyuncs.com
China (Chengdu)	cn-chengdu-intranet.log.aliyuncs.com
Hong Kong	cn-hongkong-intranet.log.aliyuncs.com
US (Silicon Valley)	us-west-1-intranet.log.aliyuncs.com

Region	Root service endpoint
Japan (Tokyo)	ap-northeast-1-intranet.log.aliyuncs.com
Singapore	ap-southeast-1-intranet.log.aliyuncs.com
Australia (Sydney)	ap-southeast-2-intranet.log.aliyuncs.com
Malaysia (Kuala Lumpur)	ap-southeast-3-intranet.log.aliyuncs.com
Indonesia (Jakarta)	ap-southeast-5-intranet.log.aliyuncs.com
UAE (Dubai)	me-east-1-intranet.log.aliyuncs.com
Germany (Frankfurt)	eu-central-1-intranet.log.aliyuncs.com
US (Virginia)	us-east-1-intranet.log.aliyuncs.com
India (Mumbai)	ap-south-1-intranet.log.aliyuncs.com
UK (London)	eu-west-1-intranet.log.aliyuncs.com

For example, if the project name is big-game and the region is China (Hangzhou), the private network service endpoint is as follows:

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

**Note:**

Currently, Log Service supports only HTTP and HTTPS for API operations through the preceding service endpoints.

Service endpoint for Internet-based Global Acceleration

In addition to VPC and the public network, Log Service adds a network type of [Internet-based Global Acceleration](#). Compared with the ordinary Internet access, Internet-based Global Acceleration has significant advantages in terms of latency and stability. It is suitable for scenarios with high requirements for data collection, low consumption latency, and reliability.

The service endpoint for Internet-based Global Acceleration is the same in all regions as follows: `log - global . aliyuncs . com`.

**Note:**

The Global Acceleration feature is disabled by default. You must manually enable it. For more information about how to enable Global Acceleration, see [#unique_62](#).

1.4 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.

1.5 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.

Header name	Type	Description
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 #unique_57 .
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.

Header name	Type	Description
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.

Header name	Type	Description
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 <code>hmac-sha1</code> is supported.
x-acss-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.

- 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 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.

1.6 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).

Header name	Type	Description
x-log-requestid	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.

1.7 Request signature

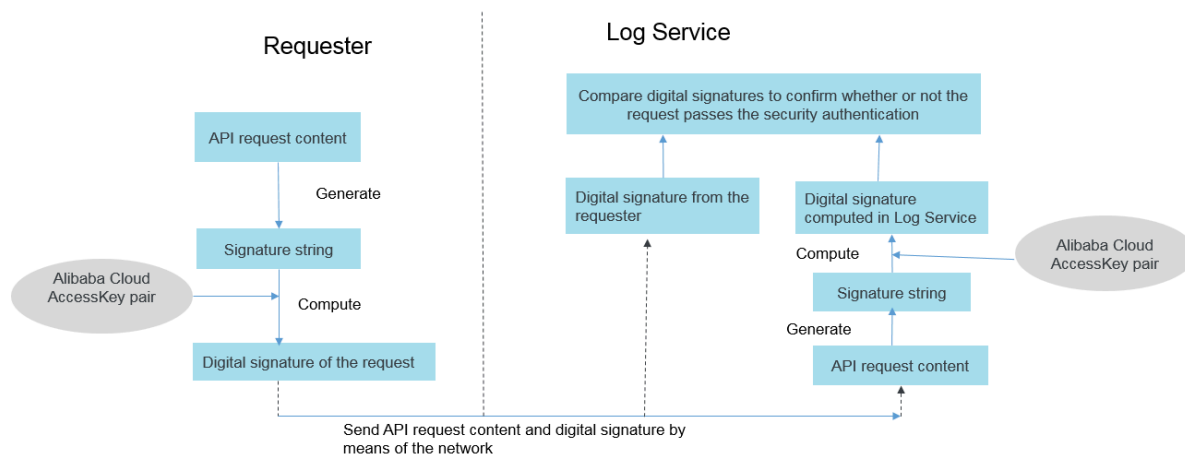
To ensure security of users' log data, all HTTP requests of the Log Service API must undergo security authentication. Currently, this security authentication is based on the Alibaba Cloud [#unique_67](#) and is completed by using the symmetric encryption algorithm.

This process is as follows:

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

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

Figure 1-1: Security verification process



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

- Confirm which user sends the API request. Because a user is required to designate the key pair to generate the digital signature before sending the request, the server can use this key pair to confirm the user's identity and then perform access permission management.
- Confirm 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**. The specific formats are 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

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. The AccessKey pair must be in the "Active" state.

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.

Table 1-1: Signing string Definitions

Name	Definitions	Example
VERB	The method name of the HTTP request.	PUT, GET, and POST
CONTENT-MD5	The type of the HTTP request body.	875264590688CA6171F6 228AF5BBB3D2
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 <code>x-log</code> and <code>x-ac</code> 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

Name	Definitions	Example
CanonicalizedResource	A string constructed by an HTTP request Resource (specific construction method to meet in detail)	/logstores/app_log

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

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

As described in [#unique_58](#), 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 ))
```

Use the HMAC-SHA1 method defined in [RFC 2104](#) as the signature method. The AccessKey Secret used in the preceding formula Service. must correspond to the AccessKey ID used in the final Authorization header. Otherwise, the request will not be able to pass authentication by the server.

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 = " bq2sjzesjm o86kq *****"
AccessKeySecret = " 4fd02fTDDn ZPU / L7CHNd *****"
```

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.regionid.example.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

```
jEY0TCJs2e88o + y5F4 / S5IsnBJQ =
```

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

```
GET / logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host : ali - test - project . regionid . example . com
x - log - apiversion : 0.6.0
x - log - signaturemethod : hmac - sha1
Authorization : LOG bq2sjzesjm o86kq35beh upbq : jEY0TCJs2e
88o + 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 = "10.10.10.1"
"TestKey": "TestContent"
```

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

```
POST / logstores / test - logstore HTTP / 1.1
Date : Mon, 09 Nov 2015 06:03:03 GMT
Host : test - project . regionid . example . com
x - log - apiversion : 0.6.0
x - log - signaturemethod : hmac - sha1
Content - MD5 : 1DD45FA4A7 0A9300CC9F E7305AF2C4 94
Content - Length : 52
x - log - apiversion : 0.6.0
x - log - bodyrawsize : 50
x - log - compress type : 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 parameters. According to the above signature string construction method, the signature string corresponding to this request is:

```
POST \ n1DD45FA4A 70A9300CC9 FE7305AF2C 494 \ napplicati on / x
- protobuf \ nMon , 09 Nov 2015 06 : 03 : 03 GMT \ nx - log
- apiversion : 0 . 6 . 0 \ nx - log - bodyrawsiz e : 50 \ nx - log -
compressty pe : lz4 \ nx - log - signaturem ethod : hmac - sha1 \ n
/ logstores / test - logstore
```

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

```
XWLGyHGg2F 2hcfxWxMLi NkGki6g =
```

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

```
POST / logstores / test - logstore HTTP / 1 . 1
Date : Mon , 09 Nov 2015 06 : 03 : 03 GMT
Host : test - project . regionid . example . com
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
Content - MD5 : 1DD45FA4A7 0A9300CC9F E7305AF2C4 94
Content - Length : 52
x - log - apiversion : 0 . 6 . 0
x - log - bodyrawsiz e : 50
x - log - compressty pe : lz4
x - log - signaturem ethod : hmac - sha1
Authorizat ion : LOG bq2sjzesjm o86kq35beh upbq : XWLGyHGg2F
2hcfxWxMLi NkGki6g =
< Log contents are serialized to byte streams in the
  ProtoBuffer format >
```

1.8 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 >,
  " errorMessa ge " : < ErrorMessa ge >
```

```
}

```

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.

Table 1-2: Common error codes

HTTP status code	Error code	Error message	Description
411	MissingContentLength	Content-Length does not exist in http header when it is necessary.	The required Content-Length request header is 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.
The HTTP standard request header Host is not provided.	Missingdate	Date does not exist in http header.	The HTTP standard request header Date is not provided.

HTTP status code	Error code	Error message	Description
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	ProjectNotExists	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.

HTTP status code	Error code	Error message	Description
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.

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

1.9 Project interfaces

1.9.1 CreateProject

Create a project.

Example:

```
POST /
```

Request syntax

```
POST / HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturem ethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / json
Content - MD5 : < Content - MD5 >
Content - Length : < ContentLen gth >
Connection : Keep - Alive
{
  " projectNam e " : < ProjectNam e >,
  " descriptio n " : < Descriptio n >
}
```

Request parameters

Attribute name	Type	Required	Description
projectName	String	Yes	Log Project name.
description	string	Yes	Project description.

Request header

The CreateProject interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The CreateProject interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	ProjectAlreadyExist	Project {project} already exist
400	ParameterInvalid	The body is not valid json string
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
POST / HTTP / 1.1
Authorization: LOG < yourAccessKeyId >:< yourSignature >
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EF F5E3CD447F B6D8DF294E 20
Content-Length: 80
Connection: Keep-Alive
{
  "projectName": "my-project-test",
  "description": "Description of my-project-test"
}
```

Response example:

```
HTTP / 1.1 200
Server: nginx
Content-Length: 0
```

```
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 27 May 2018 07 : 43 : 27 GMT
x - log - requestid : 5B0A619F20 5DC3F30EDA 9322
```

1.9.2 DeleteProject

Delete a specified project.

Example:

```
DELETE /
```

Request syntax

```
DELETE / HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturem ethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / x - protobuf
Connection : Keep - Alive
```

Request parameters

Attribute name	Type	Required or not	Description
projectName	string	Yes	Project name, which is a part of Host in the header.

Request header

The DeleteProject interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The DeleteProject interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	The Project does not exist : {Project}
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
DELETE / HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project - test . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Sun , 27 May 2018 08 : 25 : 04 GMT
Content - Type : applicatio n / x - protobuf
Connection : Keep - Alive
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 27 May 2018 08 : 25 : 04 GMT
x - log - requestid : 5B0A6B60BB 6EE39764D4 58B5
```

1.9.3 UpdateProject

Updates a Project.

Request syntax

```
PUT / HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturem ethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / json
Content - MD5 : < Content - MD5 >
Content - Length : < ContentLen gth >
```

Connection : Keep - Alive

Request parameters

Parameter	Type	Required?	Description
projectName	String	Yes	The Project name. It is part of the host field in a header.
description	String	No.	The project description. By default, this parameter is an empty string.

Request header

The UpdateProject interface does not have a special request header. For more information, see [#unique_58](#).

Response header

The UpdateProject interface does not have a special response header. For more information, see [#unique_76](#).

Response elements

The returned HTTP status code is 200.

Error codes

In addition to the [#unique_73](#) of Log Service, the following error codes that are exclusive to the UpdateProject interface may be returned.

HTTP status code	Error code	Error message	Description
404	ProjectNotExist	The Project does not exist : <Project >.	The specified Project does not exist.
400	ParameterInvalid	The body is not valid json string.	The parameter is not a valid JSON string.
500	InternalServerError	Specified Server Error Message.	An error occurred to the specified server.

Examples

Request example

```
PUT / HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project - test . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Sun , 27 May 2018 07 : 43 : 26 GMT
Content - Type : applicatio n / json
Content - MD5 : A7967D81EF F5E3CD447F B6D8DF294E 20
Content - Length : 40
Connection : Keep - Alive
{
  " descriptio n ": " Descriptio n of my - project - test "
}
```

Response example

```
HTTP / 1 . 1 200
Server : nginx
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 27 May 2018 07 : 43 : 27 GMT
x - log - requestid : 5B0A619F20 5DC3F30EDA 9322
```

1.9.4 GetProject

The GetProject interface queries a project by project name.

Example:

```
GET /
```

Request syntax

```
GET / HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturem ethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / x - protobuf
```

Connection : Keep - Alive

Request parameters:

Attribute name	Type	Required	Description
projectName	string	Yes	The project name that is a part of the host in the header.

Request header

GetProject interface does not have a specific response header. For more information about public response headers of Log Service APIs, see [#unique_58](#).

Response header

GetProject interface does not have a specific request header. For more information about public request headers of Log Service APIs, see [#unique_72](#).

Response element

If the GetProject request is sent successfully, the response body contains the detailed information of the project in the following format:

Attribute name	Type	Description
createTime	string	Creation time.
description	string	Project description.
lastModifyTime	string	Last update time.
owner	string	Account ID of the project creator.
projectName	string	Project name.
status	string	Project status.
region	string	List of excluded fields.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	The Project does not exist : {Project}

HTTP status code	ErrorCode	ErrorMessage
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
DELETE / HTTP / 1 . 1
Authorization : LOG < yourAccess KeyId >:< yourSignature >
x - log - bodyrawsize : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project - test . cn - shanghai . log . aliyuncs . com
x - log - signaturemethod : hmac - sha1
Date : Sun , 27 May 2018 08 : 25 : 04 GMT
Content - Type : application / x - protobuf
Connection : Keep - Alive
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 27 May 2018 08 : 25 : 04 GMT
x - log - requestid : 5B0A6B60BB 6EE39764D4 58B5
```

1.9.5 ListProject

Query the list of all projects.

Example

```
GET /? offset ={ offset }& size ={ size }
```

Request syntax

```
GET /? offset ={ offset }& size ={ size } HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : application / x - protobuf
```

Connection : Keep - Alive

Request parameters

Attribute name	Type	Required	Description
offset	integer	No	Start position of a returned record. The default value is 0.
size	integer	No	Maximum number of entries returned per page. The default value is 500 (maximum).

Request header

The ListProject interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The ListProject interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

If the ListProject request is sent successfully, the response body contains a project list in the following format:

Attribute name	Type	Description
count	integer	Number of projects returned.
total	integer	Total number of projects.
projects	array	Project list.

Each element of projects is shown in the following format:

Attribute name	Type	Description
createTime	string	Creation time.
description	string	Project description.
lastModifyTime	string	Last update time.

Attribute name	Type	Description
owner	string	Account ID of the project creator.
projectName	string	Project name.
status	string	Project status.
region	string	List of excluded fields.

Error code

The interface may return the following error codes in addition to Log Service API

[#unique_73](#):

HTTP status code	ErrorCode	Error message
500	InternalServerError	Specified Server Error Message

Example

Request example

```
GET /? offset = 0 & size = 2 & projectName = HTTP / 1 . 1
Authorization : LOG < yourAccessKeyId >:< yourSignature >
x-log-bodyrawsize : 0
User-Agent : sls-java-sdk-v-0.6.1
x-log-apiversion : 0.6.0
Host : cn-shanghai.log.aliyuncs.com
x-log-signaturemethod : hmac-sha1
Date : Sun , 27 May 2018 09 : 03 : 33 GMT
Content-Type : application / x-protobuf
Connection : Keep-Alive
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx
Content-Type : application / json
Content-Length : 345
Connection : close
Access-Control-Allow-Origin : *
Date : Sun , 27 May 2018 09 : 03 : 33 GMT
x-log-requestid : 5B0A7465AA EA20CA70DE 3064
{
  "count": 2 ,
  "total": 11 ,
  "projects": [
    {
      "projectName": "project1 ",
      "status": "Normal ",
      "owner": "",
      "description": "",
      "region": "cn-shanghai ",
```

```

    " createTime ": " 1524222931 ",
    " lastModify Time ": " 1524539357 "
  },
  {
    " projectName ": " project123 456 ",
    " status ": " Normal ",
    " owner ": "",
    " description ": "",
    " region ": " cn - shanghai ",
    " createTime ": " 1471963876 ",
    " lastModify Time ": " 1524539357 "
  }
]
}

```

1.9.6 GetProjectLogs

GetProjectLogs is the SQL query interface of the Project level.

Request syntax

```

GET / logs /? query = SELECT * FROM < logStoreName > 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

Attribute name	Type	Required or not	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 the returned results of the request are complete. The specific response element format is as follows.

Name	Type	Description
x-log-progress	String	The status of the query results. The two values, Incomplete and Complete , indicate whether 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.

Parameter	Type	Required or not
__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 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 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. When a portion of the query request results are cache hits, the server continues to scan the log data that are not cache hits for this request. To reduce the workload of merging multiple query results, the server cache hit query results and the new hits of the current query are merged and returned to you. Therefore, the Log Service allows you to call the interface multiple times with the same parameters to finally obtain the complete results. Log Service API cannot predict how many times the APIs 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 to continue the query. You must note that each call to this API consumes the same number of query CUs.

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.

Samples

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 < logStoreName > 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 : 36F9F7F033 9BEAF57158 1AF1B0AAAF B5
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.

1.10 Logstore APIs

1.10.1 CreateLogstore

Create a Logstore in a project.

Example:

```
POST / logstores
```

Request syntax

```
POST / logstores HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturemethod : hmac - sha1
{
  " logstoreName " : < logStoreName >,
  " ttl " : < ttl >,
  " shardCount " : < shardCount >,
  " autoSplit " : < autoSplit >,
  " maxSplitShard " : < maxSplitShard >
}
```

Request parameters

Attribute name	Type	Required	Description
logstoreName	string	Yes	The Logstore name, which must be unique in the same project.
ttl	integer	Yes	Data storage time in days, which is in the value range of 1 to 3600.
shardCount	Integer	Yes	The number of shards in this Logstore, which is in the range of 1 to 100.
enable_tracking	bool	No	Determines whether to enable Web Tracking.
autoSplit	bool	No	Determines whether to automatically split a shard.

Attribute name	Type	Required	Description
maxSplitShard	int	No	The maximum number of shards for automatic split, which is in the range of 1 to 64. You must specify this parameter when autoSplit is true.

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 [#unique_58](#).

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 [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error Code

Besides the [#unique_73](#) of Log Service APIs, the CreateLogstore API may return the following special error codes.

Table 1-3: Createlogstore-specific error codes

HTTP status code	ErrorCode	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

Details description

The Logstore cannot be created if the quota is invalid.

Examples

Request example

```
POST / logstores HTTP / 1 . 1
Header :
{
  x - log - apiversion = 0 . 6 . 0 ,
  Authorization = LOG < yourAccess KeyId >:< yourSignature >,
  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 = 7EF43D0B8F 4A807B95E7 75048C911C 72 ,
  User - Agent = sls - java - sdk - v - 0 . 6 . 0 ,
  Content - Type = application / json
}
Body :
{
  " logstoreName ": " test - logstore ",
  " ttl ": 1 ,
  " shardCount ": 2 ,
  " autoSplit ": true ,
  " maxSplitShard ": 64
}
```

Response example:

```
HTTP / 1 . 1 200 OK
Header :
{
  Date = Wed , 11 Nov 2015 07 : 35 : 00 GMT ,
  Content - Length = 0 ,
  x - log - requestid = 5642EFA499 248C827B01 2B39 ,
  Connection = close ,
  Server = nginx / 1 . 6 . 1
}
```

1.10.2 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

Table 1-4: 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 [#unique_58](#).

Response head

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 [#unique_73](#) 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 < yourAccessKeyId >:< yourSignature >,
  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
Header :
{
  Date = Wed , 11 Nov 2015 08 : 09 : 39 GMT ,
  Content - Length = 0 ,
  x - log - requestid = 5642F7C399 248C817B01 3A07 ,
  Connection = close ,
  Server = nginx / 1 . 6 . 1
}
```

1.10.3 UpdateLogstore

Update 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 > ,
  " autoSplit ": < autoSplit > ,
  " maxSplitShard ": < maxSplitShard >
}
```

Request parameters

Parameter name	Type	Required	Description
logstoreName	String	Yes	The Logstore name , which is unique in the same project.
ttl	Integer	Yes	The lifecycle (in days) of log data. The value range is 1 -3600.

Parameter name	Type	Required	Description
shardCount	Integer	Yes	Shows the number of shard. Updating Logstore attributes does not change the number of shards. You can change the number of shards through only SplisShard or MergeShard.
enable_tracking	Bool	No	Determines whether to enable Web Tracking.
autoSplit	Bool	No	Determines whether to automatically split a shard.
maxSplitShard	Int	No	The maximum number of shards for automatic split , which is in the range of 1 to 64. You must specify this parameter when autoSplit is true

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 Codes	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 modify shardCount by split&merge shard

Detailed description

Currently, the number of shards can be modified through only SplitShard or MergeShard.

Examples

Request example

```
PUT / logstores / test - logstore HTTP / 1 . 1
Header :
{
  x - log - apiversion = 0 . 6 . 0 ,
  Authorization = LOG < yourAccessKeyId >:< yourSignature >,
  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 = 757C60FC41 CC7D3F60B8 8E0D916D05 1E ,
  User - Agent = sls - java - sdk - v - 0 . 6 . 0 ,
  Content - Type = application / json
}
Body :
{
  " logstoreName ": " test - logstore ",
  " ttl ": 1 ,
  " shardCount ": 2 ,
  " autoSplit ": true ,
  " maxSplitShard ": 64
```

```
}
```

Response example:

```
HTTP / 1 . 1    200    OK
Header :
{
  Date = Wed , 11 Nov 2015 08 : 28 : 20 GMT ,
  Content - Length = 0 ,
  x - log - requestid = 5642FC2399 248C8F7B01 45FD ,
  Connection = close ,
  Server = nginx / 1 . 6 . 1
}
```

1.10.4 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 >,
    " lastModify Time ": < lastModify Time >
}

```

Error code

Besides the common error codes of Log Service APIs, the GetLogstore API may return the following special error codes. [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 ,
  Authorizat ion = LOG < yourAccess KeyId >:< yourSignat ure >,
  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 = applicatio n / json
}

```

Response example

```

HTTP / 1 . 1 200 OK
Header :
{
  Date = Wed , 11 Nov 2015 07 : 53 : 30 GMT ,
  Content - Length = 107 ,
  x - log - requestid = 5642F3FA99 248C817B01 352D ,
  Connection = close ,
  Content - Type = applicatio n / json ,
  Server = nginx / 1 . 6 . 1
}
Body :
{
  " logstoreName ": test - logstore ,
  " ttl ": 1 ,

```

```
" shardCount ": 2 ,  
" createTime ": 1447833064 ,  
" lastModify Time ": 1447833064  
}
```

1.10.5 ListLogstore

List the names of all the Logstores in a specified project.

Request syntax

```
GET / logstores HTTP / 1 . 1  
Authorizat ion : < Authorizat ionString >  
Date : < GMT Date >  
Host : < Project Endpoint >  
x - log - apiversion : 0 . 6 . 0  
x - log - signaturem ethod : hmac - sha1
```

Request parameters

Parameter Name	Type	Required	Description
offset(optional)	Integer	No	The starting position of a returned record. The default value is 0.
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 head

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	he 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	ErrorCode	ErrorMessage
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 < yourAccess KeyId >:< yourSignature >,
  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 = applicatio n / json
}
```

Response example

```
HTTP / 1 . 1 200 OK
Header :
```

```
{
  Date = Wed , 11 Nov 2015 08 : 09 : 39 GMT ,
  Content - Length = 52 ,
  x - log - requestid = 5642F7C399 248C8D7B01 342F ,
  Connection = close ,
  Content - Type = application / json ,
  Server = nginx / 1 . 6 . 1
}
Body :
{
  " count ": 1 ,
  " logstores ":[" test - logstore "],
  " total ": 1
}
```

1.10.6 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	Yes	Logstore name

Request header

No special request header is available. 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

No special response header is available. The ListShards API does not have a special request header. For more information about the public request headers of Log Service APIs, see [Public response header](#).

Response element

An array composed of shards.

```
[
  {
    " shardID ": 0 ,
    " status ": " readwrite ",
```

```

    "inclusiveBeginKey": "00000000000000000000000000000000",
    "exclusiveEndKey": "80000000000000000000000000000000",
    "createTime": 1453949705
  },
  {
    ...
  },
  {
    ...
  }
]

```

Error code

Besides the common error codes [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 <yourAccessKeyId>:<yourSignature>"
}

```

Response example

```

Header :
{

```



```

    " content - length ": " 57 ",
    " server ": " nginx / 1 . 6 . 1 ",
    " connection ": " close ",
    " date ": " Thu , 12 Nov 2015 03 : 40 : 31 GMT ",
    " content - type ": " applicatio n / json ",
    " x - log - requestid ": " 56440A2F99 248C050600 C74C "
  }
  Body :
  [
    {
      " shardID ": 1 ,
      " status ": " readwrite ",
      " inclusiveB eginKey ": " 0000000000 0000000000 0000000000
00 ",
      " exclusiveE ndKey ": " 8000000000 0000000000 0000000000 0
",
      " createTime ": 1453949705
    },
    {
      " shardID ": 2 ,
      " status ": " readwrite ",
      " inclusiveB eginKey ": " 8000000000 0000000000 0000000000
00 ",
      " exclusiveE ndKey ": " ffffffff ffffffff ffffffff
ff ",
      " createTime ": 1453949705
    },
    {
      " shardID ": 0 ,
      " status ": " readonly ",
      " inclusiveB eginKey ": " 0000000000 0000000000 0000000000
00 ",
      " exclusiveE ndKey ": " ffffffff ffffffff ffffffff
ff ",
      " createTime ": 1453949705
    }
  ]

```

1.10.7 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	Type	Required	Description
logstoreName	string	Yes	The Logstore name.
shardid	int	Yes	The shard ID.

Parameter	Type	Required	Description
Splitkey	string	Yes	The split location.

Request header

No special request header is available. For more information about the public request headers of Log Service APIs, see [Public request header](#).

Response header

Content-Type: application/json

No special response header is available. 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 ",
    " inclusiveB   eginKey ": " ee00000000  0000000000  0000000000
00 ",
    " exclusiveE   ndKey ": " ffffffff  ffffffff  ffffffff
ff ",
    " createTime ": 1453949705
  },
  {
    " shardID ": 163 ,
    " status ": " readwrite ",
    " inclusiveB   eginKey ": " ee00000000  0000000000  0000000000
00 ",
    " exclusiveE   ndKey ": " ef00000000  0000000000  0000000000
00 ",
    " createTime ": 1453949705
  },
  {
    " shardID ": 164 ,
    " status ": " readwrite ",
    " inclusiveB   eginKey ": " ef00000000  0000000000  0000000000
00 ",
    " exclusiveE   ndKey ": " ffffffff  ffffffff  ffffffff
ff ",
    " createTime ": 1453949705
  }
]
```

Error code

Besides the [common error codes](#) of Log Service APIs, the SplitShard API may return the following special error codes.

HTTP status code	ErrorCode	ErrorMessage
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 with a specific Logstore name.

Example**Request example**

```
POST / logstores / logstorename / shards / 33 ? action = split &
key = ef00000000 0000000000 0000000000 00
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 < yourAccess KeyId >:< yourSignat
ure >"
}
```

Response example

```
Header :
{
  " content - length ": " 57 ",
  " server ": " nginx / 1 . 6 . 1 ",
  " connection ": " close ",
  " date ": " Thu , 12 Nov 2015 03 : 40 : 31 GMT ",
  " content - type ": " applicatio n / json ",
  " x - log - requestid ": " 56440A2F99 248C050600 C74C "
}
Body :
[
  {
    " shardID ": 33 ,
    " status ": " readonly ",
    " inclusiveBeginKey ": " ee00000000 0000000000 0000000000
00 ",
```

```

ff ", " exclusiveE ndKey ": " ffffffff ffffffff ffffffff
    },
    {
      " shardID ": 163 ,
      " status ": " readwrite ",
      " inclusiveB eginKey ": " ee00000000 0000000000 0000000000
00 ", " exclusiveE ndKey ": " ef00000000 0000000000 0000000000
00 ", " createTime ": 1453949705
    },
    {
      " shardID ": 164 ,
      " status ": " readwrite ",
      " inclusiveB eginKey ": " ef00000000 0000000000 0000000000
00 ", " exclusiveE ndKey ": " ffffffff ffffffff ffffffff
ff ", " createTime ": 1453949705
    }
  ]

```

1.10.8 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	No	The Logstore name.
shardid	int	No	The shard ID.

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

Content-Type: application/json

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

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

```
[
  {
    ' shardID ': 167 ,
    ' status ': ' readwrite ',
    ' inclusiveB eginKey ': ' e000000000 0000000000 0000000000
00 ',
    ' createTime ': 1453953105 ,
    ' exclusiveE ndKey ': ' ffffffff ffffffff ffffffff
ff ',
  },
  {
    ' shardID ': 30 ,
    ' status ': ' readonly ',
    ' inclusiveB eginKey ': ' e000000000 0000000000 0000000000
00 ',
    ' createTime ': 0 ,
    ' exclusiveE ndKey ':
    ' e700000000 0000000000 0000000000 00 '
  },
  {
    ' shardID ': 166 ,
    ' status ': ' readonly ',
    ' inclusiveB eginKey ': ' e700000000 0000000000 0000000000
00 ',
    ' createTime ': 1453953073 ,
    ' exclusiveE ndKey ': ' ffffffff ffffffff ffffffff
ff '
  }
]
```

Error codes

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

HTTP status code	ErrorCode	ErrorMessage
404	LogStoreNotExist	logstore {logstoreName} does not exist
400	ParameterInvalid	invalid shard id
400	ParameterInvalid	can not merge the last shard

HTTP status code	ErrorCode	ErrorMessage
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.

Examples**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 < yourAccessKeyId >:< yourSignature >"
}
```

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 ": " 56440A2F99 248C050600 C74C "
}
Body :
[
  {
    ' shardID ': 167 ,
    ' status ': ' readwrite ',
    ' inclusiveBeginKey ': ' e000000000 0000000000 0000000000
00 ',
    ' createTime ': 1453953105 ,
    ' exclusiveEndKey ': ' ffffffff ffffffff ffffffff
ff '
  },
  {
    ' shardID ': 30 ,
    ' status ': ' readonly ',
    ' inclusiveBeginKey ': ' e000000000 0000000000 0000000000
00 ',
    ' createTime ': 0 ,
```

```

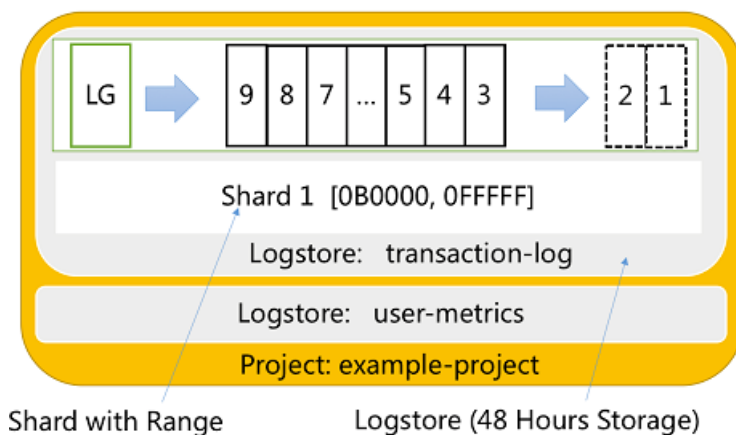
    ' exclusiveE ndKey ':
    ' e700000000 0000000000 0000000000 00 '
  },
  {
    ' shardID ': 166 ,
    ' status ': ' readonly ',
    ' inclusiveB eginKey ': ' e700000000 0000000000 0000000000
00 ',
    ' createTime ': 1453953073 ,
    ' exclusiveE ndKey ': ' ffffffff ffffffff ffffffff
ff '
  }
]

```

1.10.9 GetCursor

You can call this operation to query a cursor based on the server time. The following figure shows the relationships among the project, Logstore, shard, and cursor.

Figure 1-2: Relationships among the project, Logstore, shard, and cursor



- A project has multiple Logstores.
- A Logstore has multiple shards.
- You can use a cursor to obtain the location of a specific log.

Request syntax

```

GET / logstores / ay42 / shards / 2 ? type = cursor & from =
1402341900 HTTP / 1 . 1
Authorization : < AuthorizationString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0

```

Request parameters

Parameter	Type	Required	Description
shard	String	Yes	

Parameter	Type	Required	Description
type	String	Yes	The type of data to be queried. Set this parameter to cursor .
from	String	Yes	The time used to query a cursor. Set this parameter to a Unix timestamp or a string such as begin or end.

Logstore lifecycle

The lifecycle of a Logstore is specified by the ttl parameter in the Logstore attributes. For example, the current time is 2018-11-11 09:00:00 and the ttl parameter is set to 5. The time interval of consumable data in each shard is [2018-11-05 09:00:00, 2018-11-11 09:00:00). The server time is used.

Using the from parameter, you can locate the logs within the lifecycle of a shard. Assume that the Logstore lifecycle is [begin_time, end_time) and the from parameter is set to from_time, then:

- `from_time ≤ begin_time` or `from_time = "begin"`: returns the cursor corresponding to the log whose timestamp is at begin_time.
- `from_time ≥ end_time` or `from_time = "end"`: returns the cursor corresponding to the next log to be written at the current time (there is no data in the current cursor location).
- `from_time > begin_time` and `from_time < end_time`: returns the cursor corresponding to the first data packet that the server receives later than or at from_time.

Request header fields

This operation does not require special request header fields. For more information about the common request header fields of Log Service operations, see [Common request header fields](#).

Response header fields

This operation does not return special response header fields. For more information about the common response header fields of Log Service operations, see [Common response header fields](#).

Response parameters

```
{
  " cursor ": " MTQ0NzI5OT  YwNjg5NjYz  MjM1Ng =="
```

Error codes

In addition to the [common errors](#) of Log Service operations, this operation also returns some specific errors, as listed in the following table.

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

Examples

Sample request

```
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 ": " applicatio n / json ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >"
```

```
}
```

Sample success response

```
Header :
{
  " content - length ": " 41 ",
  " server ": " nginx / 1 . 6 . 1 ",
  " connection ": " close ",
  " date ": " Thu , 12 Nov 2015 03 : 56 : 57 GMT ",
  " content - type ": " applicatio n / json ",
  " x - log - requestid ": " 56440E0999 248C070600 C6AA "
}
Body :
{
  " cursor ": " MTQ0NzI5OT YwNjg5NjYz MjM1Ng == "
}
```

1.10.10 PullLogs

Obtains 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 =
MTQ0NzMyOT QwMTEwMjEz MDkwNA ==& count = 100 HTTP / 1 . 1
Accept : applicatio n / x - protobuf
Accept - Encoding : lz4
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

Request parameters

URL parameters:

Parameter	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.

Parameter	Type	Required	Description
count	int	Yes	The number of returned log groups , ranging from 1 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.

Error code

Besides the [common error codes](#) of Log Service APIs, the UpdateLogstore API may return the following special error codes.

HTTP status code	ErrorCode	ErrorMessage
404	LogStoreNotExist	Logstore {Name} does not exist
400	ParameterInvalid	Parameter Cursor is not valid
400	ParameterInvalid	ParameterCount should be in [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 =
MTQ0NzMyOT QwMTEwMjEz MDkwNA ==& count = 1000 & type = log
Header :
{
  " Authorization =" LOG < yourAccess KeyId >:< yourSignature
>",
  " 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 " : " applicatio n / x - protobuf ",
  " accept " : " applicatio n / x - protobuf "
}
```

Response example

```
Header :
{
  " x - log - count " : " 1000 ",
  " x - log - requestid " : " 56447FB203 51626D7C00 0874 ",
  " Server " : " nginx / 1 . 6 . 1 ",
  " x - log - bodyrawsize " : " 34121 ",
  " Connection " : " close ",
  " Content - Length " : " 4231 ",
  " x - log - cursor " : " MTQ0NzMyOT QwMTEwMjEz MDkwNA ==",
  " Date " : " Thu , 12 Nov 2015 12 : 01 : 54 GMT ",
  " x - log - compressstype " : " lz4 ",
  " Content - Type " : " applicatio n / x - protobuf "
}
Body :
The < log group list in PB format > after the
compressio n .
```

Page flip

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

1.10.11 PutLogs

You can call this operation to write log data to a specific Logstore. Currently, you can write only log groups in [Protocol Buffers \(Protobuf\)](#) format.



Note:

Log data is packaged in units of log groups.

You can write log data in either of the following modes:

- **LoadBalance mode:** Log Service automatically writes log data to all writable shards in a Logstore in load balancing mode. This mode is highly available for writing data, with the SLA at 99.95%. It is applicable to scenarios in which data is written and consumed irrelevant to shards, for example, scenarios that do not preserve the order of logs.
- **KeyHash mode:** A key is specified for writing data. Log Service automatically writes log data to the shard whose MD5 value range contains the key value. For example, Log Service can hash a producer such as an instance to a fixed shard based on the name to ensure that data is written and consumed in strict order in this shard. When shards are merged or split, a key is included only in the MD5 value range of a shard at a time. For more information, see [Shard](#).

Request syntax

LoadBalance mode

```
POST /logstores/<logstorename>/shards/lb HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/x-protobuf
Content-Length: <ContentLength>
Content-MD5: <ContentMD5>
Date: <GMTDate>
Host: <ProjectEndpoint>
x-log-apiversion: 0.6.0
x-log-bodyrawsize: <BodyRawSize>
x-log-compress-type: lz4
x-log-signaturemethod: hmac-sha1
<Compressed data of logs in Protobuf format>
```

KeyHash mode

Log Service adds the `x-log-hashkey` field to the request header to specify the shard whose MD5 value range contains the key value. This field is optional. If you do not specify this field, Log Service automatically writes log data in LoadBalance mode.

```
POST /logstores/<logstorename>/shards/lb HTTP/1.1
Authorization: <AuthorizationString>
Content-Type: application/x-protobuf
Content-Length: <ContentLength>
Content-MD5: <ContentMD5>
Date: <GMTDate>
Host: <ProjectEndpoint>
x-log-apiversion: 0.6.0
x-log-bodyrawsize: <BodyRawSize>
x-log-compress-type: lz4
x-log-hashkey: 14d2f850ad6ea48e46e4547edbbb27e0
x-log-signaturemethod: hmac-sha1
```

< Compressed data of logs in Protobuf format >

Request parameters

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

Request header fields

In KeyHash mode, the x-log-hashkey field must be added to the request header. For more information, see the preceding request syntax.

For more information about the common request header fields of Log Service operations, see [Common request header fields](#).

Response header fields

This operation does not return special response header fields. For more information about the common response header fields of Log Service operations, see [Common response header fields](#).

Response parameters

After a request is sent, no response is returned.

Detailed description

- You can use the PutLogs operation to write up to 4,096 logs, with the maximum size of 3 MB. The value of each log in a log group cannot exceed 1 MB in size. A request fails and no log data can be written if any of the preceding conditions are not met.
- Log Service checks the format of log data written by the PutLogs operation. For more information about log formats, see [Basic concepts](#). A request fails and no log data can be written if any log does not conform to the format specifications.

Error codes

In addition to the [common errors](#) of Log Service operations, this operation also returns some specific errors, as listed in the following table.

HTTP status code	Error code	Error message	Description
400	PostBodyInvalid	Protobuffer content cannot be parsed.	The error message returned because the logs in Protobuf format cannot be parsed.
400	InvalidTimestamp	Invalid timestamps are in logs.	The error message returned because the logs contain invalid timestamps.
400	InvalidEncoding	Non-UTF8 characters are in logs.	The error message returned because the logs contain non-UTF-8-encoded characters.
400	InvalidKey	Invalid keys are in logs.	The error message returned because the logs contain invalid keys.
400	PostBodyTooLarge	Logs must be less than or equal to 3 MB and 4096 entries.	The error message returned because the number of logs exceeds 4,096 or the size of logs is greater than 3 MB.
400	PostBodyUncompressError	Failed to decompress logs.	The error message returned because the logs fail to be decompressed.
499	PostBodyInvalid	The post data time is out of range	The error message returned because the log timestamp is beyond the valid interval of $[-7 \times 24 \text{ hours}, +15 \text{ minutes}]$.
404	LogStoreNotExist	logstore {Name} does not exist.	The error message returned because the specified Logstore does not exist.

**Note:**

The {name} variable in the preceding error message is replaced with a specific Logstore name.

Examples**Sample request**

```
POST / logstores / sls - test - logstore
{
  " Content - Length ": 118 ,
  " Content - Type ":" applicatio n / x - protobuf ",
  " x - log - bodyrawsiz e ": 1356 ,
  " Host ": " ali - test - project . cn - hangzhou - devcommon -
intranet . sls . aliyuncs . com ",
  " Content - MD5 ":" 6554BD0421 49C844761C 2C094A8FEC CE ",
  " Date ":" Thu , 12 Nov 2015 06 : 54 : 26 GMT ",
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " x - log - compressty pe ":" lz4 "
  " x - log - signaturem ethod ": " hmac - sha1 ",
  " Authorizat ion ":" LOG < yourAccess KeyId >:< yourSignat ure
>"
}
< Binary data obtained after logs in Protobuf format
are compressed by using the LZ4 algorithm >
```

Sample success response

```
Header
{
  " date ": " Thu , 12 Nov 2015 06 : 53 : 03 GMT ",
  " connection ": " close ",
  " x - log - requestid ": " 5644160399 248C060600 D216 ",
  " content - length ": " 0 ",
  " server ": " nginx / 1 . 6 . 1 "
}
```

1.10.12 GetShipperStatus

Query the LogShipper task status.

Request syntax

```
GET / logstores /{ logstoreNa me }/ shipper /{ shipperNam e }/
tasks ? from = 1448748198 & to = 1448948198 & status = success &
offset = 0 & size = 100 HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
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 " : " abcdefghij k ",
      " taskStatus " : " success ",
      " taskMessag e " : "",
      " taskCreate Time " : 1448925013 ,
      " taskLastDa taReceiveT ime " : 1448915013 ,
      " taskFinish Time " : 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.

Name	Type	Description
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	ErrorCode	ErrorMessage
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 support query last 48 hours task status
400	ParameterInvalid	status only contains success/running/fail

Detailed description

You can only query LogShipper task status within the last 24 hours.

Example

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 ,
  Authorizat ion = LOG < yourAccess KeyId >:< yourSignat ure >,
  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 - signaturem ethod = hmac - sha1 ,
  Content - MD5 = 757C60FC41 CC7D3F60B8 8E0D916D05 1E ,
  User - Agent = sls - java - sdk - v - 0 . 6 . 0 ,
  Content - Type = applicatio n / json
}
```

Response example

```
HTTP / 1 . 1 200 OK
```

```

Header :
{
  Date = Wed , 11 Nov 2015 08 : 28 : 20 GMT ,
  Content - Length = 0 ,
  x - log - requestid = 5642FC2399 248C8F7B01 45FD ,
  Connection = close ,
  Server = nginx / 1 . 6 . 1
}
Body :
{
  " count " : 10 ,
  " total " : 20 ,
  " statistics " : {
    " running " : 0 ,
    " success " : 20 ,
    " fail " : 0
  }
  " tasks " : [
    {
      " id " : " abcdefghij k ",
      " taskStatus " : " success ",
      " taskMessage " : "",
      " taskCreateTime " : 1448925013 ,
      " taskLastDataReceiveTime " : 1448915013 ,
      " taskFinishTime " : 1448926013
    }
  ]
}

```

1.10.13 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 head

RetryShipperTask 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	ErrorCode	ErrorMessage
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 ,
  Authorizat ion = LOG < yourAccess KeyId >:< yourSignat ure > ,
```

```
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 - signaturem ethod = hmac - sha1 ,
Content - MD5 = 757C60FC41 CC7D3F60B8 8E0D916D05 1E ,
User - Agent = sls - java - sdk - v - 0 . 6 . 0 ,
Content - Type = applicatio n / 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 = 5642FC2399 248C8F7B01 45FD ,
Connection = close ,
Server = nginx / 1 . 6 . 1
}
```

1.10.14 GetLogs

You can call this operation to query logs in a Logstore of a specific project. You can also specify relevant parameters to query logs that meet the specified conditions.

When a log is written to a Logstore, the latency of querying this log by using the GetHistograms and GetLogs operations of Log Service varies with the log type. Based on the log timestamp, Log Service classifies logs into the following types:

- **Real-time data:** Logs of this type are generated within the interval of (-180 seconds, 900 seconds] based on the current server time. For example, if a log was generated at 2014-09-25 12:03:00 UTC and the server received the log at 2014-09-25 12:05:00 UTC, the server processes the log as real-time data. Such logs are usually generated in normal scenarios.
- **Historical data:** Logs of this type are generated within the interval of $[-7 \times 86400$ seconds, -180 seconds) based on the current server time. For example, if a log was generated at 2014-09-25 12:00:00 UTC and the server received the log at 2014-09-25 12:05:00 UTC, the server processes the log as historical data. Such logs are usually generated in data supplement scenarios.

After real-time data is written, it can be queried with a maximum latency of 3 seconds . Data can be queried within 1 second in 99.9% of cases.

Request syntax

```
GET / logstores /< logstorename >? type = log & topic =< logtopic
>& from =< starttime >& to =< endtime >& query =< querystring >&
line =< linenumber >& offset =< startindex >& reverse =< true | false >
HTTP / 1 . 1
Authorization : < AuthorizationString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - bodyrawsize : 0
x - log - api version : 0 . 6 . 0
x - log - signature method : hmac - sha1
```

Request parameters

Parameter	Type	Required	Description
logstorename	String	Yes	The name of the Logstore where logs are to be queried.
type	String	Yes	The type of data to be queried in the Logstore. Set this parameter to log for the GetLogs operation.
from	Integer	Yes	The start time of queried data, in seconds. Set this parameter to the number of seconds that have elapsed since 1970-1-1 00:00:00 UTC.
to	Integer	Yes	The end time of queried data, in seconds. Set this parameter to the number of seconds that have elapsed since 1970-1-1 00:00:00 UTC.
topic	String	No	The topic of logs to be queried.

Parameter	Type	Required	Description
query	String	No	The query statement. For more information about the query syntax, see #unique_98 .
line	Integer	No	The maximum number of logs to return for the request. Valid values: [0, 100]. Default value: 100.
offset	Integer	No	The start point of the specified time interval for the request. Valid values: 0 and positive integers. Default value: 0.
reverse	Boolean	No	Specifies whether to return logs in reverse order based on the log timestamp, in minutes. A value of true indicates that logs are returned in reverse order. A value of false indicates that logs are returned in regular order. Default value: false.

Request header fields

The GetLogs operation does not require special request header fields. For more information about the common request header fields of Log Service operations, see [Common request header fields](#).

Response header fields

For more information about the common response header fields of Log Service operations, see [Common response header fields](#).

The response header contains specific fields that indicate whether the query results returned for a request are complete. The following table describes these response header fields.

Field	Type	Description
x-log-progress	String	The status of the query results. The value can be Incomplete or Complete, which indicates whether the results are complete.
x-log-count	Integer	The total number of logs in the current query results.

Response parameters

After a request is sent for the GetLogs operation, the response body contains logs that meet the specified query conditions. If a large number of logs (in units of TB) need to be queried, the response of the GetLogs operation may be incomplete. The response body of the GetLogs operation is an array, and each element in this array is a log. The following table describes the structure of each element in the array.

Parameter	Type	Description
__time__	Integer	The timestamp of the log, in seconds. The value indicates the number of seconds that have elapsed since 1970-1-1 00:00:00 UTC.
__source__	String	The source of the log, which is specified when the log is 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 for this operation is left-closed and right-open. That is, the time interval includes the start time but excludes the end time. If the values of the from and to parameters are the same, the time interval is invalid and the server directly returns an error.
- The server must return a response to each request for this operation within a specified time. For each query, the server can scan only a specified number of logs. If the server needs to process a large number of logs for a request, it may return incomplete query results. Whether the results are complete is indicated by the x-log-progress field in the response header. At the same time, the server caches the query results within 15 minutes. If some query results of a request hit the data in the cache, the server continues to scan the logs that do not hit any data in the cache for this request. To reduce your workload of merging multiple query results, the server merges the query results that hit the data in the cache with the results newly scanned in the current query, and then returns the merged results to you. In this manner, Log Service allows you to call the GetLogs operation multiple times with the same parameters to obtain the final complete results. Log Service cannot predict how many times it must call the GetLogs operation before obtaining the complete results because the volume of logs to be queried changes dramatically. Therefore, you must check the value of the x-log-progress field in the returned results of each request to determine whether to continue the query. You must note that each call of this operation consumes the same number of compute units (CUs).

Error codes

In addition to the [common errors](#) of Log Service operations, this operation also returns some specific errors, as listed in the following table.

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

HTTP status code	Error code	Error message	Description
400	InvalidQueryString	query string is invalid	The error message returned because the specified query string of the request is invalid.
400	InvalidOffset	offset is invalid	The error message returned because the specified offset parameter of the request is invalid.
400	InvalidLine	line is invalid	The error message returned because the specified line parameter of the request is invalid.
400	InvalidReverse	Reverse value is invalid	The error message returned because the specified reverse parameter is invalid.
400	IndexConfigNotExist	logstore without index config	The error message returned because the index feature has not been enabled for the Logstore.

**Note:**

The {name} variable in the preceding error message is replaced with a specific Logstore name.

Examples

Take a project named big-game in China (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 to 2014-09-01 22:00:00, and the query keyword is error. The query starts from the start time of the time interval, and a maximum of 20 logs are returned.

Sample request

```
GET / logstores / app_log ? type = log & topic = groupA & from =
1409529600 & to = 1409608800 & query = error & line = 20 & offset = 0
HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : Wed , 3 Sept . 2014 08 : 33 : 46 GMT
Host : big - game . cn - hangzhou . log . aliyuncs . com
x - log - bodyrawsiz e : 0
x - log - apiversion : 0 . 4 . 0
x - log - signaturem ethod : hmac - sha1
```

Sample success response

```
HTTP / 1 . 1 200 OK
Content - MD5 : 36F9F7F033 9BEAF57158 1AF1B0AAAF B5
Content - Type : applicatio n / 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 - millisecon d : 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 sample response, the value of the x-log-progress field is Complete, which indicates that the log query is completed and the returned results are complete. For this request, two logs meet the specified query conditions and are displayed as the value of the logs parameter. If the value of the x-log-progress field is Incomplete in the response, you must repeat the request to obtain the complete results.

1.10.15 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.

Parameter name	Type	Required	Description
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 #unique_98 .

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
Authorizat ion : < Authorizat ionString >
Date : Wed , 3 Sept . 2014 08 : 33 : 46 GMT
Host : big - game . cn - hangzhou . log . aliyuncs . com
x - log - bodyrawsiz e : 0
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

Response example

```
HTTP / 1 . 1 200 OK
Content - Type : applicatio n / json
Content - MD5 : E6AD9C2120 4868C2DE84 EE3808AAA8 C8
Content - Type : applicatio n / 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 : applicatio n / json
Content - MD5 : E6AD9C2120 4868C2DE84 EE3808AAA8 C8
Content - Type : applicatio n / 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 "
    }
  ]
}
```

1.10.16 CreateIndex

Create an index for a specified Logstore.

Example:

```
POST / logstores /{ logstoreName }/ index
```

Request syntax

```
POST / logstores /< logstoreName >/ index HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : < x - log - bodyrawsiz e >
User - Agent : < User - Agent >
x - log - apiversion : < APIVersion >
Host : < Project Endpoint >
x - log - signaturem ethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / json
Content - MD5 : < Content - MD5 >
Content - Length : < Content - Length >
{
  " line ": < full text index > ,
```

```
" keys ": < key - value    index >
}
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Logstore name.
keys	object	No	Index configuration of the key value. Key indicates the field name, and value indicates the index configuration.
line	object	No	Full text index configuration.

Specify either keys or line. The full text index configuration includes the following attributes:

Attribute name	Type	Required or not	Description
caseSensitive	bool	No	Case sensitivity.
chn	bool	No	Include Chinese characters or not.
token	array	Yes	Word divider list.
include_keys	array	No	List of included fields, and the attribute cannot be specified with exclude_keys at the same time.
exclude_keys	array	No	List of excluded fields, and the attribute cannot be specified with exclude_keys at the same time.

The field index configuration includes the following attributes:

Attribute name	Type	Required	Description
type	string	Yes	Field type.
alias	string	No	Field alias.
chn	bool	No	Include Chinese characters or not . This field makes sense only when the value of type is text.
token	array	This field is required only when the value of type is text.	Word divider list . This field makes sense only when the value of type is text.
caseSensitive	bool	No	Case sensitivity. This field makes sense only when the value of type is text.
doc_value	bool	No	Enable or disable statistics analysis on the field.

Request header

The CreateIndex interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The CreateIndex interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#) .

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	IndexInfoInvalid	required field token is lacking or of error format
400	IndexAlreadyExist	log store index is already created
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

**Note:**

In the error message in the above table, {Project} and {logstoreName} indicate that this section will be replaced by a specific project name and Logstore name.

Example**Request example**

```
POST / logstores / my - logstore / index HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Mon , 07 May 2018 09 : 43 : 16 GMT
Content - Type : applicatio n / json
Content - MD5 : 22876515FC 311F857AD6 C7902F6A01 48
Content - Length : 316
Connection : Keep - Alive
{
  " line ": {
    " token ": [
      ",",
      ",",
      ",",
      ",",
      "\"",
      ";",
      "=",
      "(",
      ")",
      "[",
      "]",
      "{",
      "}",
      "?",
      "@",
      "&",
      "<"
    ]
  }
}
```

```

    ">",
    "/" ,
    ":" ,
    "\ n " ,
    "\ t " ,
    "\ r "
  ]
},
" keys ": {
  " agent ": {
    " doc_value ": true ,
    " caseSensitive ": true ,
    " alias ": " agent_aliases " ,
    " type ": " text " ,
    " token ": [
      " , " ,
      " " ,
      " " ,
      " \" " ,
      " ; " ,
      " = " ,
      " ( " ,
      " ) " ,
      " [ " ,
      " ] " ,
      " { " ,
      " } " ,
      " ? " ,
      " @ " ,
      " & " ,
      " < " ,
      " > " ,
      " / " ,
      " : " ,
      "\ n " ,
      "\ t " ,
      "\ r "
    ]
  }
}
}
}

```

1.10.17 DeleteIndex

Deletes an index for a specified Logstore.

Example:

```
DELETE /logstores/{logstoreName}/index
```

Request syntax

```
DELETE /logstores /< logstoreName >/ index HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
```

```
Content - Type : application / x - protobuf
Connection : Keep - Alive
```

Request parameters:

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Logstore name.

Request header

The DeleteIndex interface does not have a specific request header. For more information, see [#unique_58](#).

Response header

The DeleteIndex interface does not have a specific response header. For more information, see [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	Error code	Error message
400	ParameterInvalid	log store index not created
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

Example

Request example

```
DELETE / logstores / my - logstore / index HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
```



```
Date : Sun , 06 May 2018 13 : 22 : 57 GMT
Content - Type : application / x - protobuf
Connection : Keep - Alive
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 06 May 2018 13 : 22 : 57 GMT
x - log - requestid : 5AEF01B1A4 58E41C2F83 26A8
```

1.10.18 GetIndex

Query the index of a specified Logstore.

Example:

```
GET / logstores /{ logstoreName }/ index
```

Request syntax

```
GET / logstores /< logstoreName >/ index HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : application / x - protobuf
Connection : Keep - Alive
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Logstore name.

Request header

The GetIndex interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The GetIndex interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

When the GetIndex request is sent successfully, the response body contains the indexes of a specified project and Logstore in the following format:

Attribute name	Type	Required or not
index_mode	string	Index type.
keys	dict	Index configuration of the key value. Key indicates the field name, and value indicates the index configuration.
line	object	Full text index configuration.
storage	string	Storage type, which is pg at present.
ttl	integer	Index file lifecycle. The lifecycle can be 7 days, 30 days or 90 days.
lastModifyTime	integer	Last update time of the index in the format of Unix timestamp.

The full text index configuration includes the following attributes:

Attribute name	Type	Required or not
caseSensitive	bool	Case sensitivity.
chn	bool	Include Chinese characters or not.
token	array	Word divider list.
include_keys	array	List of included fields.
exclude_keys	array	List of excluded fields.

Configuration corresponding to each field in the key value index contains the following attributes:

Attribute name	Type	Required or not
type	string	Field type.
alias	string	Field alias.

Attribute name	Type	Required or not
chn	bool	Include Chinese characters or not. The field is available only when the value of type is text.
token	array	Word divider list. The field is available only when the value of type is text.
caseSensitive	bool	Case sensitivity. The field is available only when the value of type is text.
doc_value	bool	Enable or disable statistics analysis on the field.

Error code

The interface may return the following error codes in addition to Log Service API

#unique_73:

HTTP status code	ErrorCode	ErrorMessage
400	IndexConfigNotExist	index config doesn' t exist
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

Example

Request example

```
GET / logstores / logstore - 4 / index HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Sun , 06 May 2018 13 : 08 : 42 GMT
Content - Type : applicatio n / x - protobuf
```

Connection : Keep - Alive

Response example

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Type : applicatio n / json
Content - Length : 712
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 06 May 2018 13 : 08 : 42 GMT
x - log - requestid : 5AEFE5A8B 8AEB5E6C82 B395
{
  " index_mode ": " v2 ",
  " keys ": {
    " agent ": {
      " alias ": "",
      " caseSensit ive ": false ,
      " chn ": false ,
      " doc_value ": true ,
      " token ": [
        ",",
        " ",
        "!",
        "\"",
        "\",
        ";",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "&",
        "<",
        ">",
        "/",
        ":",
        "\ n ",
        "\ t ",
        "\ r "
      ],
      " type ": " text "
    },
    " bytes ": {
      " alias ": "",
      " doc_value ": true ,
      " type ": " long "
    },
    " remote_ip ": {
      " alias ": "",
      " caseSensit ive ": false ,
      " chn ": false ,
      " doc_value ": true ,
      " token ": [
        ",",
        " ",
        "!",
        "\"",
        "\",
        ";",
        "=",

```

```

        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "&",
        "<",
        ">",
        "/",
        ":",
        "\\ n ",
        "\\ t ",
        "\\ r "
    ],
    "type ": " text "
},
"response ": {
    "alias ": "",
    "doc_value ": true ,
    "type ": " long "
}
},
"line ": {
    "caseSensit ive ": false ,
    "chn ": false ,
    "token ": [
        ",",
        " ",
        "'",
        "\"",
        "\\\"",
        ";",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "&",
        "<",
        ">",
        "/",
        ":",
        "\\ n ",
        "\\ t ",
        "\\ r "
    ]
},
"storage ": " pg ",
"ttl ": 30 ,
"lastModify Time ": 1524155379
}

```

1.10.19 UpdateIndex

Update an index for a specified Logstore.

Example:

```
PUT / logstores /{ logstoreName }/ index
```

Request syntax

```
PUT / logstores /< logstoreName >/ index HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / json
Content - MD5 : < Content - MD5 >
Content - Length : < ContentLength >
{
  " line ": < full text index >,
  " keys ": < key - value index >
}
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Logstore name.
keys	object	No	Index configuration of the key value. Key indicates the field name, and value indicates the index configuration.
line	object	No	Full text index configuration.

Specify either keys or line. The full text index configuration includes the following attributes:

Attribute name	Type	Required or not	Description
token	array	Yes	Word divider list.
caseSensitive	bool	No	Case sensitivity.
chn	bool	No	Include Chinese characters or not.

Attribute name	Type	Required or not	Description
include_keys	array	No	List of included fields, and the attribute cannot be specified with excluded_keys at the same time.
exclude_keys	array	No	List of excluded fields, and the attribute cannot be specified with excluded_keys at the same time.

The field index configuration includes the following attributes:

Attribute name	Type	Required or not	Description
type	string	Yes	Field type.
alias	string	No	Alias.
chn	bool	No	Include Chinese characters or not . This field makes sense only when the value of type is text.
token	array	No. This field is required when the value of type is text.	Word divider list . This field makes sense only when the value of type is text.
caseSensitive	bool	No	Case sensitivity. This field makes sense only when the value of type is text.
doc_value	bool	No	Enable statistics or not.

Request header

The UpdateIndex interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

UpdateIndex interface does not have a specific response header. For more information about public request headers of Log Service APIs, see [#unique_72](#).

Response element

The system returns the HTTP status code 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	ParameterInvalid	log store index is not created
400	ParameterInvalid	key config must has type
400	IndexInfoInvalid	required field token is lacking or of error format
400	IndexInfoInvalid	required fields line/keys are lacking or of error format
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

Example

· Request example

```
PUT / logstores / logstore - 4 / index HTTP / 1 . 1
Header :
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Mon , 07 May 2018 09 : 47 : 20 GMT
```



```

Content - Type : application / json
Content - MD5 : 1860B805A6 AA5288B97B 32CF3B5191 12
Content - Length : 316
Connection : Keep - Alive
Body :
{
  "line": {
    "token": [
      ",",
      "\"",
      "'",
      "\"",
      "\\",
      "\"",
      "\"",
      "=",
      "(",
      ")",
      "[",
      "]",
      "{",
      "}",
      "?",
      "@",
      "&",
      "<",
      ">",
      "/",
      ":",
      "\n",
      "\t",
      "\r"
    ]
  },
  "keys": {
    "agent": {
      "doc_value": true,
      "caseSensitive": true,
      "alias": "agent_aliases",
      "type": "text",
      "token": [
        ",",
        "\"",
        "'",
        "\"",
        "\\",
        "\"",
        "\"",
        "=",
        "(",
        ")",
        "[",
        "]",
        "{",
        "}",
        "?",
        "@",
        "&",
        "<",
        ">",
        "/",
        ":",
        "\n",
        "\t",
        "\r"
      ]
    }
  }
}

```

```
}
```

- **Response example**

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Mon , 07 May 2018 09 : 47 : 21 GMT
x - log - requestid : 5AF020A98C BAA2EF52AB 66C9
```

1.11 Logtail machine group related interfaces

1.11.1 Createmachinegroup

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

Request syntax

```
POST / machinegro ups HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Content - Type : applicatio n / json
Content - Length :< Content Length >
Content - MD5 <:< Content MD5 >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
{
  " groupName " : " testgroup ",
  " groupType " : "",
  " groupAttri bute " : {
    " externalNa me " : " testgroup ",
    " groupTopic " : " testgroup topic "
  },
  " machineIde ntifyType " : " ip ",
  " machineLis t " : [
    " 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.

Parameter name	Type	Required	Description
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	ErrorCode	ErrorMessage
400	MachineGroupAlreadyExist	group {GroupName} already exists
400	InvalidParameter	invalid group resource json
500	InternalServerError	Internal server error

Example

Request example

```
POST /machinegroups HTTP/1.1
Header :
{
  "x-log-apiversion": "0.6.0",
  "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
  "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 ": " 5642300D99 248CB76D00 5D36 ",
  " Connection ": " close ",
  " Server ": " nginx / 1 . 6 . 1 "
}
```

1.11.2 DeleteMachineGroup

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

Example:

```
DELETE / machinegro ups /{ groupName }
```

Request syntax

```
DELETE / machinegro ups /{ groupName } HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : 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

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

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

Example

Request example

```
DELETE / machinegroup / test - machine - group - 4 HTTP / 1 . 1
Header :
{
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " Authorization ": " LOG < yourAccessKeyId >: < yourSignature >",
  " 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 ": " 564241D899 248C827B00 0CFE ",
  " Connection ": " close ",
  " Server ": " nginx / 1 . 6 . 1 "
}
```

1.11.3 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 / machinegroupups /{ groupName }
```

Request syntax

```
PUT / machinegroupups /{ 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 " : " testgroup topic "
  },
  " 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:

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.

Attribute name	Type	Required	Description
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	ErrorCode	ErrorMessage
404	GroupNotExist	group {GroupName} does not exist
400	InvalidParameter	invalid group resource json
500	InternalServerError	internal server error

Example

Request example

```
PUT / machinegroup / test - machine - group HTTP / 1 . 1
Header :
{
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " Authorization ": " LOG < yourAccessKeyId >:< yourSignature >",
  " 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 ": " 2CEBAEBE53 C078891527 CB70A855BA F4 ",
  " 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 ": " 56423A6799 248CA57B00 035C ",
" Connection ": " close ",
" Server ": " nginx / 1 . 6 . 1 "
}
```

1.11.4 ListMachineGroup

View the list of machine group names.

Example:

```
GET / machinegro ups ? offset = 1 & size = 100
```

Request syntax

```
GET / machinegro ups ? offset = 1 & size = 100 HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

Request Parameters

URL parameters:

Parameter Name	Type	Required	Description
offset	int	No	The starting position of the returned records. The default value is 0.
size	int	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 The specific formats are as follows.

Name	Type	Description
count	int	Number of machinegroup returned
total	int	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	ErrorCode	ErrorMessage
500	InternalServerError	internal server error

Example

Request example:

```
GET / machinegro ups ? groupName = test - machine - group & offset
= 0 & size = 3 HTTP / 1 . 1
Header :
{
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >",
  " 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 - signaturem ethod ": " hmac - sha1 ",
  " User - Agent ": " sls - java - sdk - v - 0 . 6 . 0 ",
  " Content - Type ": " applicatio n / x - protobuf ",
  " x - log - bodyrawsiz e ": " 0 "
}
```

Response example:

```
HTTP / 1 . 1 200 OK
Header :
{
  " Date ": " Tue , 10 Nov 2015 18 : 34 : 44 GMT ",
  " Content - Length ": " 83 ",
  " x - log - requestid ": " 564238C499 248C8F7B00 01DE ",
  " Connection ": " close ",
  " Content - Type ": " applicatio n / json ",
  " Server ": " nginx / 1 . 6 . 1 "
}
Body :
{
  " machinegro ups ": [
    " test - machine - group ",
    " test - machine - group - 2 "
  ],
  " count ": 2 ,
  " total ": 2
}
```

1.11.5 GetMachineGroup

View details of a machine group.

Example:

```
GET / machinegro ups /{ groupName }
```

Request syntax

```
GET / machinegro ups /{ groupName } HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
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 group type (empty or Armory), which is empty by default.
groupAttribute	4) JSON object	The machine identification type, including IP and user-defined identity.
machineList	json array	The specific machine identification, which can be an IP address or user-defined identity.
createTime	int	The created time of the machine group.

Attribute name	Type	Description
lastModifyTime	int	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. External Management System (armory) Identification on which Machine grouping is dependent

```
{
  " 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

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

Example

Request example:

```
GET / machinegro ups / test - machine - group HTTP / 1 . 1
Header :
{
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >",
  " 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 - signaturem ethod ": " hmac - sha1 ",
  " User - Agent ": " sls - java - sdk - v - 0 . 6 . 0 ",
  " Content - Type ": " applicatio n / x - protobuf ",
  " x - log - bodyrawsiz e ": " 0 "
}
```

Response example:

```
HTTP / 1 . 1 200 OK
Header :
{
  " Date ": " Tue , 10 Nov 2015 18 : 15 : 23 GMT ",
  " Content - Length ": " 239 ",
  " x - log - requestid ": " 5642343B99 248CB36D00 60B8 ",
  " Connection ": " close ",
  " Content - Type ": " applicatio n / json ",
  " Server ": " nginx / 1 . 6 . 1 "
}
Body :
{
  " groupName ": " test - machine - group ",
  " groupType ": "",
  " groupAttri bute ": {
    " externalNa me ": " testgroup ",
    " groupTopic ": " testtopic "
  },
  " machineIde ntifyType ": " ip ",
  " machineLis t ": [
    "\" 127 . 0 . 0 . 1 \"",
    "\" 127 . 0 . 0 . 2 \"",
  ],
  " createTime ": 1447178253 ,
  " lastModify Time ": 1447178253
}
```

1.11.6 ApplyConfigToMachineGroup

Apply a configuration to a machine group.

Example:

```
PUT / machinegro ups /{ GroupName }/ configs /{ ConfigName }
```

Request syntax

```
GET / machinegro ups /{ groupName } HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

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	ErrorCode	ErrorMessage
404	GroupNotExist	group {GroupName} does not exist

HTTP status code	ErrorCode	ErrorMessage
404	ConfigNotExist	config {ConfigName} does not exist
500	InternalServerError	internal server error

Example

Request example:

```
PUT / machinegro ups / sample - group / configs / logtail - config
- sample
Header :
{
  " Content - Length ": 0 ,
  " x - log - signaturem ethod ": " hmac - sha1 ",
  " x - log - bodyrawsiz e ": 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 ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >"
}
```

Response example:

```
{
  " date ": " Mon , 09 Nov 2015 09 : 44 : 43 GMT ",
  " connection ": " close ",
  " x - log - requestid ": " 56406B0B99 248CAA230B A094 ",
  " content - length ": " 0 ",
  " server ": " nginx / 1 . 6 . 1 "
}
```

1.11.7 RemoveConfigFromMachineGroup

Remove a configuration from a machine group.

Example:

```
DELETE / machinegro ups /{ GroupName }/ configs /{ ConfigName }
```

Request syntax

```
GET / machinegro ups /{ groupName } HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
```

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

Request parameters

Parameter Name	Type	Required or not	Description
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	ErrorMessage
404	GroupNotExist	group {GroupName} does not exist
404	ConfigNotExist	config {ConfigName} does not exist
500	InternalServerError	internal server error

Example

Request example:

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

```
{
  " Content - Length ": 0 ,
  " x - log - signaturem ethod ": " hmac - sha1 ",
  " x - log - bodyrawsiz e ": 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 ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >"
}
```

Response example:

```
{
  " date ": " Mon , 09 Nov 2015 09 : 48 : 48 GMT ",
  " connection ": " close ",
  " x - log - requestid ": " 56406C0099 248CAA230B E135 ",
  " content - length ": " 0 ",
  " server ": " nginx / 1 . 6 . 1 "
}
```

1.11.8 ListMachines

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

Example:

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

Request syntax

```
GET / machinegro ups /{ groupName }/ machines HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

Request Parameters

URL parameters:

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

Name	Type	Required	Description
offset	int	No	The starting position of the returned records. The default value is 0.
size	int	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	Int	The number of returned machines.
total	int	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.

Name	Type	Description
userdefined-id	String	The user-defined identity of the machine.

```
{
  " count ": 10 ,
  " total ": 100 ,
  " machines ":
  [{
    " ip " : " testip1 ",
    " machine - uuid " : " testuuid1 ",
    " userdefine d - id " : " testuserde finedid1 ",
    " lastHeartbeatTime " : 1447182247
  },
  {
    " ip " : " testip1 ",
    " machine - uuid " : " testuuid2 ",
    " userdefine d - id " : " testuserde finedid2 ",
    " lastHeartbeatTime " : 1447182247
  },
  {
    " ip " : " testip2 ",
    " machine - uuid " : " testuuid ",
    " userdefine d - id " : " testuserde finedid "
    " 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	ErrorCode	Error Message
404	GroupNotExist	group {GroupName} does not exist
500	InternalServerError	internal server error



Note:

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

Example

Request example:

```
GET / machinegroup / test - machine - group - 5 / machines ?
offset = 0 & size = 3 HTTP / 1 . 1
Header :
{
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " Authorization ": " LOG < yourAccessKeyId >: < yourSignature >",
```

```

    " 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 - signaturem ethod ": " hmac - sha1 ",
    " User - Agent ": " sls - java - sdk - v - 0 . 6 . 0 ",
    " Content - Type ": " applicatio n / x - protobuf ",
    " x - log - bodyrawsiz e ": " 0 "
  }
}

```

Response example:

```

HTTP / 1 . 1 200 OK
Header :
{
  " Date ": " Tue , 10 Nov 2015 19 : 04 : 58 GMT ",
  " Content - Length ": " 324 ",
  " x - log - requestid ": " 56423FD999 248C827B00 0A57 ",
  " Connection ": " close ",
  " Content - Type ": " applicatio n / json ",
  " Server ": " nginx / 1 . 6 . 1 "
}
Body :
{
  " machines ": [
    {
      " ip ": " 10 . 101 . 166 . 116 ",
      " machine - uniqueid ": "",
      " userdefine d - id ": "",
      " lastHeartb eatTime ": 1447182247
    },
    {
      " ip ": " 10 . 101 . 165 . 193 ",
      " machine - uniqueid ": "",
      " userdefine d - id ": "",
      " lastHeartb eatTime ": 1447182246
    },
    {
      " ip ": " 10 . 101 . 166 . 91 ",
      " machine - uniqueid ": "",
      " userdefine d - id ": "",
      " lastHeartb eatTime ": 1447182248
    }
  ],
  " count ": 3 ,
  " total ": 8
}

```

1.11.9 GetAppliedConfigs

Obtain the name of the configuration applied to a machine group.

Example:

```
GET / machinegro ups /{ groupName }/ configs
```

Request syntax

```
GET / machinegro ups /{ 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	Machine grouping 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. 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	ErrorCode	ErrorMessage
404	GroupNotExist	group {GroupName} does not exist
500	InternalServerError	internal server error

Example

Request example

```
GET / machinegro ups / test - machine - group / configs HTTP / 1
. 1
Header :
{
  " x - log - apiversion ": " 0 . 6 . 0 ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >",
  " 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 - signaturem ethod ": " hmac - sha1 ",
  " User - Agent ": " sls - java - sdk - v - 0 . 6 . 0 ",
  " Content - Type ": " applicatio n / x - protobuf ",
  " x - log - bodyrawsiz e ": " 0 "
}
```

Response example

```
HTTP / 1 . 1 200 OK
Header :
{
  " Date ": " Tue , 10 Nov 2015 19 : 45 : 48 GMT ",
  " Content - Length ": " 53 ",
  " x - log - requestid ": " 5642496C99 248C8C7B00 173F ",
  " Connection ": " close ",
  " Content - Type ": " applicatio n / json ",
  " Server ": " nginx / 1 . 6 . 1 "
}
Body :
{
  " configs ": [
    " two ",
    " three ",
    " test_logst ore "
  ],
  " count ": 3
}
```

1.12 Logtail configuration related interfaces

1.12.1 CreateConfig

Create a Logtail configuration in a project.

Example:

```
POST / configs
```

Request syntax

```
POST / configs HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Content - Type : applicatio n / json
Content - Length :< Content Length >
Content - MD5 <:< Content MD5 >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
{
  " configName ": " testcatego ry1 ",
  " inputType ": " file ",
  " inputDetail ": {
    " logType ": " common_reg _log ",
    " logPath ": "/ var / log / httpd /",
    " filePatter n ": " access *. log ",
    " localStora ge ": true ,
    " timeFormat ": "% Y /% m /% d % H :% M :% S ",
    " logBeginRe gex ": ".*",
    " regex ": "(\ w +)(\ s +)",
    " key " :[" key1 ", " key2 "],
    " filterKey " :[" key1 "],
    " filterRege x " :[" regex1 "],
    " fileEncodi ng ":" utf8 ",
    " topicForma t ": " none "
  },
  " outputType ": " LogService ",
  " outputDetail ":
  {
    " logstoreNa me ": " perfcounte r "
  }
}
```

Request Parameters

Parameter 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.

Parameter name	Type	Required	Description
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	YesYes.	The pattern of the log file, such as 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.

Attribute name	Type	Required	Description
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.

Attribute name	Type	Required	Description
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, <code>// var / log / (. *) . log</code>. • <code>none</code> indicates the topic is empty. • <code>default</code> indicates to use the log file path as the topic. • <code>group_topic</code> indicates to use the topic attribute of the machine group that applies this configuration as the topic.
preserve	boolean	No	<code>true</code> indicates that the monitored directory never times out. <code>false</code> indicates that the timeout for monitored directory is 30 minutes. The default value is <code>true</code>.

Attribute name	Type	Required	Description
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	ErrorCode	ErrorMessage
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 - bodyrawsiz e ': 737 ,
  ' Content - MD5 ': ' FBA01ECF72 55BE143379 BC70C56BBF 68 ',
  ' x - log - signaturem ethod ': ' 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 ': ' applicatio n / json ',
  ' Authorizat ion ': ' LOG < yourAccess KeyId >:< yourSignat
ure >'
}
Body :
{
  " configName ": " sample - logtail - config ",
  " inputType ": " file ",
  " inputDeta il ": {
    " logType ": " common_reg _log ",
    " logPath ": "/ var / log / httpd /",
    " filePatter n ": " access *. log ",
    " localStora ge ": true ,
    " timeFormat ": "% d /% b /% Y :% H :% M :% S ",
    " logBeginRe gex ": "\\ d +\\.\\.\\. d +\\.\\.\\. d +\\.\\.\\. d +\\.\\.\\. d +
- .*",
    " regex ": "([\\ d \\.] +) \\ S + \\ S + \\[(\\ S +) \\ S +\\
\\] \\\"(\\ w +) ([^\" ]*)\\\" ([\\ d \\.] +) (\\ d +) (\\ d +) (\\ d +|-)
\\\"([^\" ]*)\\\" \\\"([^\" ]*)\\\".*",
    " key ": [ " ip ", " time ", " method ", " url ", " request_ti
me ", " request_le ngth ", " status ", " length ", " ref_url ", "
browser "],
    " filterKey ": [],
    " filterRege x ": [],
    " topicForma t ": " none ",
    " fileEncodi ng ": " utf8 "
  },
  " outputType ": " LogService ",
  " outputDeta il ":
  {
    " logstoreNa me ": " 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 ': ' 56404F1A99 248CA26C00 2180 ',
  ' content - length ': ' 0 ',
  ' server ': ' nginx / 1 . 6 . 1 '
}
```

1.12.2 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
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

Request parameters

URL parameters

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.
integer	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	ErrorCode	ErrorMessage
404	ConfigNotExist	config {Configname} does not exist
500	InternalServerError	internal server error

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 < yourAccessKeyId >:< yourSignature >"
}
```



```
}
```

Response example

```
Header :
{
  " content - length ": " 103 ",
  " server ": " nginx / 1 . 6 . 1 ",
  " connection ": " close ",
  " date ": " Mon , 09 Nov 2015 09 : 19 : 13 GMT ",
  " content - type ": " applicatio n / json ",
  " x - log - requestid ": " 5640651199 248CAA2300 C2BA "
}
Body :
{
  " count ": 3 ,
  " configs ":
  [
    " logtail - config - sample ",
    " logtail - config - sample - 2 ",
    " logtail - config - sample - 3 "
  ],
  " total ": 3
}
```

1.12.3 GetAppliedMachineGroups

List the machines that apply the configuration.

Example:

```
GET / configs /{ configName }/ machinegro ups
```

Request syntax

```
GET / configs /{ configName }/ machinegro ups HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
```

Request Parameters

URL parameters:

Parameter Name	Type	Required	Description
ConfigName	string	Yes	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 List, in the following format:

Name	Type	Description
count	Integer	The number of returned machine groups. The number of machinegroup returned.
machinegroups	String Array	The name list of returned machine groups.

```
{
  " count ": 2 ,
  " machinegro ups ":
    [" 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

Example

Request example:

```
GET / configs / logtail - config - sample / machinegro ups
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 < yourAccess KeyId >:< yourSignature >"
}

```

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 ": " 56406CAA99 248CAA230B E828 "
}
Body :
{
  " count ": 1 ,
  " machinegroups ":
  [
    " sample - group "
  ]
}

```

1.12.4 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	Int	The created time of the configuration.
lastModifyTime	Int	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 /.

Attribute name	Type	Description
filePattern	String	The pattern of a log file. For example, <code>access *.log</code> .
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, <code>%Y/%m/%d %H:%M:%S</code> .
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.

Attribute name	Type	Description
topicFormat	string	Use a part of the log file path as the topic. For example, <code>/ var / log / (. *) . log</code> . 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

HTTP status code	Error Code	Error Message
500	InternalServerError	Specified Server Error Message

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 < yourAccessKeyId >:< yourSignature >"
}
```

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 ": " 5640595B99 248CAA2300 4A59 "
}
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 ,
" lastModify Time ": 1447050456
}

```

1.12.5 DeleteConfig

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

Example:

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

Request syntax

```

DELETE / configs /< configName > HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturemethod : hmac - sha1

```

Request parameters

URL parameters:

Parameter	Type	Required	Description
ConfigName	String	Yes	The configuration name.

Request header

No special request header is available. For more information about the public request headers of Log Service APIs, see [Public request header](#).

Response header

No special response header is available. For more information about the public response headers of Log Service APIs, see [Public response header](#).

Response element

The system returns the HTTP status code 200.

Error codes

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

HTTP status code	ErrorCode	ErrorMessage
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 < yourAccessKeyId >:< yourSignature >"
}
```

Response example

```
Header :
{
  " date ": " Mon , 09 Nov 2015 09 : 28 : 21 GMT ",
  " connection ": " close ",
  " x - log - requestid ": " 5640673599 248CAA2308 36C6 ",
  " content - length ": " 0 ",
  " server ": " nginx / 1 . 6 . 1 "
}
```

1.12.6 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
Authorizat ion : < Authorizat ionString >
Content - Type : applicatio n / json
Content - Length :< Content Length >
Content - MD5 <:< Content MD5 >
Date : < GMT Date >
Host : < Project Endpoint >
x - log - apiversion : 0 . 6 . 0
x - log - signaturem ethod : hmac - sha1
{
  " configName ": " testcatego ry1 ",
  " inputType ": " file ",
  " inputDetail ": {
    " logType ": " common_reg _log ",
    " logPath ": "/ var / log / httpd /",
    " filePatter n ": " access . log ",
    " localStora ge ": true ,
    " timeFormat ": "% Y /% m /% d % H :% M :% S ",
    " logBeginRe gex ": ".*",
    " regex ": "(\ w +)(\ s +)",
    " key " :[" key1 ", " key2 "],
    " filterKey " :[" key1 "],
    " filterRege x " :[" regex1 "],
    " topicForma t ": " none "
  },
  " outputType ": " LogService ",
  " outputDetail ":
  {
    " logstoreNa me ": " perfcounte r "
  }
}
```

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	The output type . Currently, only LogService is supported.

Attribute name	Type	Required	Description
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.

Attribute name	Type	Required	Description
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, <code>/ var / log / (. *) . log .</code> . The default value is none, which indicates the topic is empty.

Attribute name	Type	Required	Description
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. The UpdateConfig API does not have a special 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 ": " 431263EB10 5D584A5555 762A81E869 C0 ",
  " x - log - signature method ": " 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 ": " applicatio n / json ",
  " Authorizat ion ": " LOG < yourAccess KeyId >:< yourSignat
ure >"
}
Body :
{
  " outputDetail ": {
    " logstoreName ": " sls - test - logstore "
  },
  " inputType ": " file ",
  " inputDetail ": {
    " regex ": " ([\\ d \\.] +) \\ S + \\ S + \\ [(\\ S +) \\ S + \\
\\ \" (\\ w +) ([^\" ]*) \\ \" ([\\ 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": "564063F899 248CAA2300 B778",
  "content-length": "0",
  "server": "nginx / 1 . 6 . 1 "
}

```

1.13 Consumer group interfaces

1.13.1 CreateConsumerGroup

Creates a consumer group in a specified Logstore.

Example:

```
POST /logstores/{logstoreName}/consumergroups
```

Request syntax

```

POST /logstores/<logstoreName>/consumergroups HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <ProjectEndpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMTDate>
Content-Type: application/json
Content-MD5: F58544E4D0 22CC28A93D 0B7CC208A5 AA

```

```
Content - Length : < ContentLength >
{
  " consumerGroup ": < consumerGroup >,
  " timeout ": < timeout >,
  " order ": < order >
}
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Logstore name.
consumerGroup	string	Yes	Consumer group name, which must be unique under a project.
timeout	integer	Yes	Timeout period. If no heartbeat is received in the timeout period, the consumer group is deleted.

Attribute name	Type	Required or not	Description
order	bool	Yes	Set the system whether to consume the data of a shard in order. · true: indicates to set the system to consume the data of a shard in order. Specifically, after a shard is split to two new shards, the system firstly consumes the data in the original shard, then consumes the data in the two new shards at the same time. · false: indicates to set the system not to consume the data of a shard in order.

Request header

The CreateConsumerGroup interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The CreateConsumerGroup interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	ConsumerGroupAlreadyExist	consumer group already exist
400	JsonInfoInvalid	consumerGroup or timeout is of error format
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
POST / logstores / my - logstore / consumergr oups HTTP / 1 . 1
Header :
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Fri , 04 May 2018 08 : 02 : 22 GMT
Content - Type : applicatio n / json
Content - MD5 : F58544E4D0 22CC28A93D 0B7CC208A5 AA
Content - Length : 65
Connection : Keep - Alive
Body :
{
  " consumerGr oup " : " consumer - group - 1 ",
  " timeout " : 300 ,
  " order " : true
}
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Fri , 04 May 2018 08 : 15 : 11 GMT
x - log - requestid : 5AEC168FA7 96F4195BF4 04CB
```

1.13.2 DeleteConsumerGroup

Delete a specified consumer group.

Example:

```
DELETE / logstores /{ logstoreName }/ consumergroups /{
consumerGroupName }
```

Request syntax

```
DELETE / logstores /< logstoreName >/ consumergroups /<
consumerGroupName > HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : application / x - protobuf
Content - MD5 : F58544E4D0 22CC28A93D 0B7CC208A5 AA
```

Request parameters

Attribute name	Type	Required	Description
logstoreName	string	Yes	Name of the Logstore of the consumer group.
consumerGroup	string	Yes	Name of the deleted consumer group.

Request header

The DeleteConsumerGroup interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The DeleteConsumerGroup interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

Detailed description

If you have deleted a consumer group that does not exist, success is returned.

Example

Request example:

```
DELETE / logstores / logstore - test / consumergr oups / consumer -
group - 1 HTTP / 1 . 1
Header :
Authorizat ion : LOG :< yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Fri , 04 May 2018 08 : 02 : 22 GMT
Content - Type : applicatio n / json
Content - MD5 : F58544E4D0 22CC28A93D 0B7CC208A5 AA
Content - Length : 65
Connection : Keep - Alive
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Fri , 04 May 2018 08 : 15 : 11 GMT
x - log - requestid : 5AEC168FA7 96F4195BF4 04CB
```

1.13.3 GetCheckPoint

Obtain the checkpoints of one or all shards consumed by a specified consumer group.

Example:

```
GET / logstores /{ logstoreName }/ consumergroups /{ consumerGroup
oupName }? shard ={ shardId }
```

Request syntax

```
GET / logstores /< logstoreName >/ consumergroups /< consumerGroup
oup >? shard =< shardId > HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : application / json
Connection : Keep - Alive
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Name of the Logstore of the consumer group.
consumerGroup	string	Yes	Consumer group name.
shardId	integer	No	The shard ID. If the Shard ID is not specified, checkpoints of all shards are returned.

Request header

The GetCheckpoint interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The GetCheckpoint interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

If the GetCheckPoint request is sent successfully, the response body contains the checkpoint of the shard consumed by the specified consumer group in the following format:

Attribute name	Type	Description
shard	integer	The shard ID.
checkpoint	string	Checkpoint value.
updateTime	integer	Last update time of the checkpoint.
consumer	string	Consumer of the shard.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
404	ConsumerGroupNotExist	consumer group not exist
500	InternalServerError	Specified Server Error Message

Detailed description

If the specified shard does not exist, an empty list is returned. If the shard ID is not specified, checkpoints of all shards are returned.

Example

Request example:

```
GET / logstores / my - logstore / consumergr oups / consumer_g
roup_test ? shard = 0
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Fri , 04 May 2018 09 : 26 : 53 GMT
Content - Type : applicatio n / x - protobuf
```

Connection : Keep - Alive

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Type : applicatio n / json
Content - Length : 111
Connection : close
Access - Control - Allow - Origin : *
Date : Fri , 04 May 2018 09 : 26 : 53 GMT
x - log - requestid : 5AEC275D1F FC036AB254 B20C
[
  {
    " shard ": 0 ,
    " checkpoint ": " MTUyNDE1NT M30TM3Mzkw ODQ5Ng ==",
    " updateTime ": 1524224984 800922 ,
    " consumer ": " consumer_1 "
  }
]
```

1.13.4 HeartBeat

Send a heartbeat packet to the server for a specified consumer.

Example:

```
POST / logstores /{ logstoreName }/ consumergroups /{
consumerGroupName }? type = heartbeat & consumer ={ consumer }
```

Request syntax

```
POST / logstores /< logstoreName >/ consumergroups /<
consumerGroupName >? type = heartbeat & consumer =< consumer > HTTP /
1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / json
Content - MD5 : F58544E4D0 22CC28A93D 0B7CC208A5 AA
Content - Length : < ContentLength >
< shard ID list >
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	The Logstore name , which must be unique in the same project.

Attribute name	Type	Required or not	Description
consumerGroup	string	Yes	Consumer group name, which must be unique under a project.
consumer	string	Yes	Consumer.
{Shard ID List}	array	Yes	ID list of shards being consumed.

Request header

The HeartBeat interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The HeartBeat interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

If a HeartBeat request is sent successfully, the HTTP status code 200 and the ID list of all shards consumed by the consumer are returned.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	NotExistConsumerWith Body	non-exist consumer with non-empty body of heartbeat message
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
404	ConsumerGroupNotExist	consumer group not exist
500	InternalServerError	Specified Server Error Message

Detailed description

The heartbeat packet is sent only when the connection between the consumer and the server is established.

Example

Request example

```
POST / logstores / my - logstore / consumergr oups / consumer_g
roup_test ? type = heartbeat & consumer = consumer_1 HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Sun , 06 May 2018 09 : 44 : 10 GMT
Content - Type : applicatio n / json
Content - MD5 : 8D5162CA10 4FA7E79FE8 0FD92BB657 FB
Content - Length : 3
Connection : Keep - Alive
[ 0 ]
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Type : applicatio n / json
Content - Length : 5
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 06 May 2018 09 : 44 : 11 GMT
x - log - requestid : 5AECE6B1F FC0366B255 3FB5
[ 0 , 1 ]
```

1.13.5 ListConsumerGroup

Query all consumer groups of a specified Logstore.

Example:

```
GET / logstores /{ logstoreNa me }/ consumergr oups
```

Request syntax

```
GET / logstores /< logstoreNa me >/ consumergr oups HTTP / 1 .
1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturem ethod : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / x - protobuf
```

Connection : Keep - Alive

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Logstore name.

Request header

The ListConsumerGroup interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The ListConsumerGroup interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

If the ListConsumerGroup request is sent successfully, the response body contains a list of all consumer groups under the specified project and Logstore in the following format:

Attribute name	Type	Description
Consumergroup	string	Consumer group name.
timeout	integer	Timeout period. If no heartbeat is received in the timeout period, the consumer group is deleted.
order	bool	Sequential consumption or not.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist

HTTP status code	ErrorCode	ErrorMessage
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
GET / logstores / my - logstore / consumergr oups HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Fri , 04 May 2018 08 : 47 : 30 GMT
Content - Type : applicatio n / x - protobuf
Connection : Keep - Alive
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Type : applicatio n / json
Connection : close
Access - Control - Allow - Origin : *
Date : Fri , 04 May 2018 08 : 47 : 31 GMT
x - log - requestid : 5AEC1E2304 8191954B42 EAB9
[
  {
    " name ": " test - consumer - group ",
    " timeout ": 30 ,
    " order ": true
  }
]
```

1.13.6 UpdateCheckPoint

The UpdateCheckpoint interface updates the checkpoints of a shard of consumer groups under a specified project and Logstore.

Example:

```
POST / logstores /{ logstoreNa me }/ consumergr oups /{
consumerGr oupName }? forceSucce ss ={ forceSucce ss }& type =
checkpoint & consumer ={ consumer }
```

Request syntax

```
POST / logstores /< logstoreNa me >/ consumergr oups /<
consumerGr oupName >? forceSucce ss =< forceSucce ss >& type =
checkpoint & consumer =< consumer > HTTP / 1 . 1
Authorizat ion : < Authorizat ionString >
x - log - bodyrawsiz e : 0
User - Agent : < UserAgent >
```

```
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signature method : hmac - sha1
Date : < GMT Date >
Content - Type : applicatio n / json
Content - Length : < ContentLen gth >
Connection : Keep - Alive
{
  " shard ": < shardID >,
  " checkpoint ": < checkpoint >
}
```

Request parameters

Name	Parameters	Required or not	Description
logstoreName	string	Yes	Name of the Logstore of the consumer group.
consumerGroup	string	Yes	Consumer group name.
forceSuccess	bool	No	Force update or not .
consumer	string	No	Consumer.
shard	integer	Yes	The shard ID.
checkpoint	string	Yes	Checkpoint value.

Request header

The UpdateCheckpoint interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The UpdateCheckpoint interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

The system returns the HTTP status code 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	InvalidShardCheckPoint	shard checkpoint not encoded by base64
404	ProjectNotExist	The Project does not exist : {Project}
404	LogStoreNotExist	logstore {logstoreName} dose not exist
404	ConsumerGroupNotExist	consumer group not exist
404	ShardNotExist	shard not exist
500	InternalServerError	Specified Server Error Message

Detailed description

If the consumer is not specified, checkpoints can be updated only when forceSuccess is set to true.

Example

Request example:

```
POST / logstores / logstore - 4 / consumergr oups / consumer_g
roup_test ? forceSucce ss = false & type = checkpoint & consumer =
consumer_1 HTTP / 1 . 1
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Sun , 06 May 2018 09 : 14 : 53 GMT
Content - Type : applicatio n / json
Content - MD5 : 59457BAFB3 F85A5117BD E243947583 B0
Content - Length : 55
Connection : Keep - Alive
{
  " shard ": 0 ,
  " checkpoint ": " MTUyNDE1NT M30TM3Mzkw ODQ5Ng =="
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Sun , 06 May 2018 09 : 14 : 54 GMT
```

```
x - log - requestid : 5AEEC78E1F FC0366B24F 1C63
```

1.13.7 UpdateConsumerGroup

Modifies the attributes of a specified consumer group.

Example:

```
PUT / logstores /{ logstoreName }/ consumergroups /{ consumerGroupName }
```

Request syntax

```
PUT / logstores /< logstoreName >/ consumergroups /< consumerGroupName > HTTP / 1 . 1
Authorization : < AuthorizationString >
x - log - bodyrawsize : 0
User - Agent : < UserAgent >
x - log - apiversion : 0 . 6 . 0
Host : < Project Endpoint >
x - log - signaturemethod : hmac - sha1
Date : < GMT Date >
Content - Type : application / json
Content - MD5 : F58544E4D0 22CC28A93D 0B7CC208A5 AA
Content - Length : < ContentLength >
{
  " timeout ": < timeout >,
  " order ": < order >
}
```

Request parameters

Attribute name	Type	Required or not	Description
logstoreName	string	Yes	Name of the Logstore of the consumer group.
consumerGroup	string	Yes	Consumer group name.
timeout	integer	No	Timeout period. If no heartbeat is received in the timeout period, the consumer group is deleted.

Attribute name	Type	Required or not	Description
order	bool	No	Set the system whether to consume the data of a shard in order. · true: indicates to set the system to consume the data of a shard in order. Specifically, after a shard is split to two new shards, the system firstly consumes the data in the original shard, then consumes the data in the two new shards at the same time. · false: indicates to set the system not to consume the data of a shard in order.

**Note:**

Specify either `timeout` or `order` .

Request header

The UpdateConsumerGroup interface does not have a specific request header. For details about public request headers of Log Service APIs, see [#unique_58](#).

Response header

The UpdateConsumerGroup interface does not have a specific response header. For details about public response headers of Log Service APIs, see [#unique_72](#).

Response element

The returned HTTP status code is 200.

Error code

The interface may return the following error codes in addition to Log Service API [#unique_73](#):

HTTP status code	ErrorCode	ErrorMessage
400	JsonInfoInvalid	timeout is of error value type
404	ProjectNotExist	The Project does not exist : {Project}
404	ConsumerGroupNotExist	consumer group not exist
404	LogStoreNotExist	logstore {logstoreName} dose not exist
500	InternalServerError	Specified Server Error Message

Example

Request example:

```
PUT / logstores / logstore - test / consumergr oups / consumer -
group - 1 HTTP / 1 . 1
Header :
Authorizat ion : LOG < yourAccess KeyId >:< yourSignat ure >
x - log - bodyrawsiz e : 0
User - Agent : sls - java - sdk - v - 0 . 6 . 1
x - log - apiversion : 0 . 6 . 0
Host : my - project . cn - shanghai . log . aliyuncs . com
x - log - signaturem ethod : hmac - sha1
Date : Fri , 04 May 2018 08 : 02 : 22 GMT
Content - Type : applicatio n / json
Content - MD5 : F58544E4D0 22CC28A93D 0B7CC208A5 AA
Content - Length : 65
Connection : Keep - Alive
Body :
{
  " order ": false ,
  " timeout ": 100
}
```

Response example:

```
HTTP / 1 . 1 200
Server : nginx / 1 . 12 . 1
Content - Length : 0
Connection : close
Access - Control - Allow - Origin : *
Date : Fri , 04 May 2018 08 : 15 : 11 GMT
```



```
x - log - requestid : 5AEC168FA7 96F4195BF4 04CB
```

1.14 External Storage

1.14.1 External Store related APIs

Create an External Store

Request syntax

```
POST /externalstores HTTP/1.1
' x - log - bodyrawsize ': ' 0 ',
' Content - Type ': ' application/json ',
' Content - Length ': < ContentLength >,
' Content - MD5 ': < Md5 >,
' x - log - apiversion ': ' 0 . 6 . 0 ',
' x - log - signaturemethod ': ' hmac - sha1 ',
' Host ': < Project Endpoint >
' Date ': < GMT Date >,
' Authorization ': < AuthorizationString >
{" externalStoreName ": "< ExternalStoreName >",
  " storeType ": " rds - vpc ",
  " parameter ": {
    " vpc - id ": "< vpc - id >",
    " instance - id ": "< instance - id >",
    " host ": "< host >",
    " port ": "< port >",
    " username ": "< username >",
    " password ": "< password >",
    " db ": "< db >",
    " table ": "< table >",
    " region ": " region >"
  }
}'
```

Request Parameters

Attribute name	Type	Required or not	Description
ExternalStoreName	String	Yes	The External Store name, which is unique in a project and does not conflict with the Logstore name.
vpc-id	String	No	The Virtual Private Cloud (VPC) ID of the RDS instance.

Attribute name	Type	Required or not	Description
instance-id	String	No	The RDS instance ID. vpc-id and instance-id can be empty at the same time, which indicates the External Store is not in the VPC environment and is directly accessible.
host	String	No	The host in which the RDS instance resides, which is required if both vpc-id and instance-id are left empty.
port	String	Yes	The RDS port.
username	String	Yes	The name of the user.
password	String	Yes	The password.
db	String	Yes	Name of a database.
Table	String	Yes	The table name.
region	String	Yes	The region in which the RDS instance resides, which currently only supports cn-qingdao, cn-beijing, and cn-hangzhou.

Request Header

The CreateExternalStore API does not have a special request header. For more information about the public request headers of Log Service APIs, see [Public request header](#).

Example

```
POST /externalstores HTTP/1.1
'x-log-bodyrawsize': '0',
'Content-Type': 'application/json',
```

```
' Content - Length ': ' 307 ',
' Content - MD5 ': ' 7C1D14659C 0BBBA7C7BF F9E5A1A467 05 ',
' x - log - apiversion ': ' 0 . 6 . 0 ',
' x - log - signaturemethod ': ' hmac - sha1 ',
' Host ': ' ali - yunlei - chengdu . cn - chengdu . log . aliyuncs .
com ',
' Date ': ' Thu , 19 Apr 2018 02 : 15 : 41 GMT ',
' Authorization ': ' LOG < yourAccess KeyId >:< yourSignature
>'}
{" externalStoreName ": " rds_store ",
" storeType ": " rds - vpc ",
" parameter ": {
    " vpc - id ": " vpc -*****",
    " instance - id ": " i -*****",
    " host ": "*.*.*.*",
    " port ": " 3306 ",
    " username ": " root ",
    " password ": " sfdsfldsfk sflsdfs ",
    " db ": " meta ",
    " table ": " join_meta ",
    " region ": " cn - qingdao "
  }}'
```

Update an External Store Store

Request syntax

```
PUT / externalstores /< ExternalStoreName > HTTP / 1 . 1
' x - log - bodyrawsize ': ' 0 ',
' Content - Type ': ' application / json ',
' Content - Length ': < ContentLength >,
' Content - MD5 ': < Md5 >,
' x - log - apiversion ': ' 0 . 6 . 0 ',
' x - log - signaturemethod ': ' hmac - sha1 ',
' Host ': < Project Endpoint >
' Date ': < GMT Date >,
' Authorization ': < AuthorizationString >
{" externalStoreName ": "< ExternalStoreName >",
" storeType ": " rds - vpc ",
" parameter ": {
    " vpc - id ": "< vpc - id >",
    " instance - id ": "< instance - id >",
    " host ": "< host >",
    " port ": "< port >",
    " username ": "< username >",
    " password ": "< password >",
    " db ": "< db >",
    " table ": "< table >",
    " region ": " region >"
  }
}'
```

Request Parameters

Attribute name	Type	Required or not	Description
ExternalStoreName	string	Yes	The External Store name, which is unique in a project and does not conflict with the Logstore name.
vpc-id	String	No	The Virtual Private Cloud (VPC) ID of the RDS instance.
instance-id	String	No	The RDS instance ID. vpc-id and instance-id can be empty at the same time, which indicates the External Store is not in the VPC environment and is directly accessible.
host	String	No	The host in which the RDS instance resides, which is required if both vpc-id and instance-id are left empty.
port	String	Yes	The RDS port.
username	String	Yes	The name of the user.
password	String	Yes	The password.
db	String	Yes	Name of a database.
Table	String	Yes	The table name.
region	string	Yes	The region in which the RDS instance resides, which currently only supports cn-qingdao, cn-beijing, and cn-hangzhou.

Sample

```
PUT http://ali-yunlei-chengdu.cn-chengdu.log.aliyuncs
.com:80/externalstores/rds_store HTTP/1.1
' x-log-bodyrawsize ': ' 0 ',
' Content-Type ': ' application/json ',
' Content-Length ': ' 307 ',
' Content-MD5 ': ' 7C1D14659C 0BBBA7C7BF F9E5A1A467 05 ',
' x-log-apiversion ': ' 0.6.0 ',
' x-log-signaturemethod ': ' hmac-sha1 ',
' Host ': ' ali-yunlei-chengdu.cn-chengdu.log.aliyuncs
.com ',
' Date ': ' Thu, 19 Apr 2018 02:15:41 GMT ',
' Authorization ': ' LOG <yourAccessKeyId>:<yourSignature>'
'{"externalStoreName": "rds_store",
  "storeType": "rds-vpc",
  "parameter": {
    "vpc-id": "vpc-*****",
    "instance-id": "i-*****",
    "host": "*.*.*.*",
    "port": "3306",
    "username": "root",
    "password": "sfdsfldsfk sflsdfs",
    "db": "meta",
    "table": "join_meta",
    "region": "cn-qingdao"
  }}'
```

List all External Stores

Request syntax

```
GET /externalstores?externalStoreName=<external_s
tore_name_prefix>&offset=<offset>&lines=<lines>
' Content-Length ': ' 0 ',
' x-log-bodyrawsize ': ' 0 ',
' x-log-apiversion ': ' 0.6.0 ',
' x-log-signaturemethod ':
' hmac-sha1 ',
' Host ': '<endpoint>',
' Date ': ' Thu, 19 Apr 2018 03:03:16 GMT ',
' Authorization ': ' LOG <yourAccessKeyId>:<yourSignature>'
'>'}
```

Request Parameters

Attribute name	Type	Required or not	Description
externalStoreName	String	No	Filter the External Stores containing this string.
The value range is 0–100 and the default value is 100.	Integer	No	The offset from which you start to obtain External Stores.

Attribute name	Type	Required or not	Description
lines	Integer	No	The number of External Stores to be obtained.

Example

```
GET http://ali-yunlei-chengdu.cn-chengdu.log.aliyuncs.com:80/externalstores?externalStoreName=&offset=0&lines=10
' Content-Length ': ' 0 ',
' x-log-bodyrawsize ': ' 0 ',
' x-log-apiversion ': ' 0.6.0 ',
' x-log-signaturemethod ':
' hmac-sha1 ',
' Host ': ' ali-yunlei-chengdu.cn-chengdu.log.aliyuncs.com ',
' Date ': ' Thu, 19 Apr 2018 03:03:16 GMT ',
' Authorization ': ' LOG <yourAccessKeyId>:<yourSignature>' }
```

Response

```
{ 'count': 3, 'externalstores': ['ecs_store', 'rds_store', 'ecs_store1'], 'total': 3 }
```

Obtain External Store details

```
http://ali-yunlei-chengdu.cn-chengdu.log.aliyuncs.com:80/externalstores/<external_store_name>
' Content-Length ': ' 0 ',
' x-log-bodyrawsize ': ' 0 ',
' x-log-apiversion ': ' 0.6.0 ',
' x-log-signaturemethod ': ' hmac-sha1 ',
' Host ': ' ali-yunlei-chengdu.cn-chengdu.log.aliyuncs.com ',
' Date ': ' Thu, 19 Apr 2018 03:26:49 GMT ',
' Authorization ': ' LOG <yourAccessKeyId>:<yourSignature>' }
```

Request Parameters

Attribute name	Type	Required or not	Description
external_store_name	String	Yes	The External Store name to be obtained.

Example

```
http://ali-yunlei-chengdu.cn-chengdu.log.aliyuncs.com:80/externalstores/rds_store
' Content-Length ': ' 0 ',
' x-log-bodyrawsize ': ' 0 ',
' x-log-apiversion ': ' 0.6.0 ',
```

```
' x - log - signaturemethod ': ' hmac - sha1 ',
' Host ': ' ali - yunlei - chengdu . cn - chengdu . log . aliyuncs .
com ',
' Date ': ' Thu , 19 Apr 2018 03 : 26 : 49 GMT ',
' Authorization ': ' LOG < yourAccess KeyId >:< yourSignature >'
```

Response

```
{
  ' storeType ': ' rds - vpc ',
  ' parameter ': {
    ' region ': ' cn - qingdao ',
    ' vpc - id ': ' vpc -*****',
    ' instance - id ': ' i -*****',
    ' host ': '*.*.*.*',
    ' port ': ' 3306 ',
    ' username ': ' root ',
    ' db ': ' meta ',
    ' table ': ' join_meta '
  }
}
```

Delete an External Store

```
DELETE / externalstores /< external_store_name >
' Content - Length ': ' 0 ',
' x - log - bodyrawsize ': ' 0 ',
' x - log - apiversion ': ' 0 . 6 . 0 ',
' x - log - signaturemethod ': ' hmac - sha1 ',
' Host ': ' ali - yunlei - chengdu . cn - chengdu . log . aliyuncs .
com ',
' Date ': ' Thu , 19 Apr 2018 03 : 32 : 49 GMT ',
' Authorization ': ' LOG < yourAccess KeyId >:< yourSignature >'
```

Request Parameters

Attribute name	Type	Required or not	Description
external_store_name	string	Yes	The External Store name to be deleted.

Example

```
DELETE http:// ali - yunlei - chengdu . cn - chengdu . log .
aliyuncs . com : 80 / externalstores / rds_store
' Content - Length ': ' 0 ',
' x - log - bodyrawsize ': ' 0 ',
' x - log - apiversion ': ' 0 . 6 . 0 ',
' x - log - signaturemethod ': ' hmac - sha1 ',
' Host ': ' ali - yunlei - chengdu . cn - chengdu . log . aliyuncs .
com ',
' Date ': ' Thu , 19 Apr 2018 03 : 32 : 49 GMT ',
```

```
' Authorization ': ' LOG < yourAccess KeyId >:< yourSignature >'
```

1.14.2 External Store overview

Differences between External Store and Logstore

The Logstore stores logs and continuously receives incremental data.

The External Store stores static metadata associated with logs, and is called the external data source. Log Service usually performs join query between logs and other related data.

Scenarios

- Logstore and External Store perform join query and analysis.
- Log Service writes the query and analysis results of Logstore to External Store.

Supported storage types

Currently, only RDS instances in the Virtual Private Cloud (VPC) environment in cn-beijing, cn-qingdao, and cn-hangzhou regions can be used as External Stores. To use RDS instances in other regions as External Stores, open a ticket.

Procedure

1. Create an External Store and specify the corresponding metadata.
2. When querying and analyzing logs, reference the External Store name to query and write logs.

For more information about creating, updating, and deleting External Stores, see [#unique_132](#) related APIs or Python SDK [#unique_133](#).

1.15 RAM/STS

1.15.1 Overview

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

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

However, in scenarios where a primary account has a sub-account, you cannot use an unauthorized sub-account to perform operations on resources of the primary

account. You must grant the permission to the sub-account to perform operations on resources of the primary account using RAM authorization.

**Note:**

Before using RAM to grant the sub-account permissions to access Log Service resources of a primary account, make sure that you have carefully read [#unique_136](#) and [#unique_137](#).

Authorization policies for Log Service are available on RAM console:

- **AliyunLogFullAccess**

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

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

- **AliyunLogReadOnlyAccess**

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

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

```
}
```

- Upload data to the specified Logstore

This policy is used to grant the sub-account permission to upload data directly to the specified Logstore by using the API/SDK. The authorization policy is described as follows:

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [
        " log : Post *",
        " log : BatchPost *"
      ],
      " Resource ": [" acs : log :*: project /< specific
proejct   name >/ logstore /< specific   logstore   name >"],
      " Effect ": " Allow "
    }
  ]
}
```

- Query data of a specific Logstore in the console

This policy is used to grant the sub-account permission to have read-only access to the specified Logstore resources of the primary account (view and extract logs, and view the Logstore list). The authorization policy is described as follows:

```
{
  " Version ": " 1 ",
  " Statement ": [
    {
      " Action ": [" log : ListProjec  t ", " log : ListLogSto
res "],
      " Resource ": [" acs : log :*: project /< specific
project   name >/ *"],
      " Effect ": " Allow "
    },
    {
      " Action ": [" log : Get *"],
      " Resource ": [" acs : log :*: project /< specific
proejct   name >/ logstore /< specific   logstore   name >"],
      " Effect ": " Allow "
    }
  ]
}
```

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

For more information:

- [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](#)

1.15.2 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}: \$ {maid }: project/\$ {project name}/logstore /*
Project/Config	acs:log:\${regionName}:\${projectOwnerAliUid}:project/\${projectName}/logtailconfig/\${logtailconfig}
	ACS: Log: \$ {regionname}: \$ {maid }: project/\$ {project name}/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}

Resource type	Description method in authorization policy
	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.

See the table below for a description of each parameter.

Table 1-5:

Parameters	Description
<code>\${ regionName }</code>	indicates the name of a region.
<code>\${ projectOwn erAliUid }</code>	indicates your Alibaba Cloud account ID.
<code>\${ projectName }</code>	indicates the name of a Log Service project.
<code>\$ { LogstoreName }</code>	indicates the name of a Logstore.
<code>\${ logtailcon fig }</code>	indicates the name of a configuration.
<code>\${ machineGro upName }</code>	indicates the name of a machine group.
<code>\${ shipperNam e }</code>	indicates the name of a log shipping rule.
<code>\${ consumerGr oupName }</code>	indicates the name of a consumer group.
<code>\${ savedSearc hName }</code>	indicates the name of a saved search.
<code>\${ dashboardN ame }</code>	indicates the name of a dashboard.
<code>\${ alarmName }</code>	indicates the name of an alarm rule.

1.15.3 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.

Action	Description
log:CreateProject	Create a project.
log:GetLogStoreLogs	Query logs in a Logstore of a specific project.
log:GetLogStoreHistogram	Query the distribution of logs in a Logstore of a specified project.
log:GetLogStore	View Logstore attributes.
log:ListLogStores	List names of all Logstore of a specified project.
log:CreateLogStore	Create a Logstore in a project.
log>DeleteLogStore	Delete a Logstore, including all shards and indexes in the Logstore.

Action	Description
log:UpdateLogStore	Update Logstore attributes.
log:GetCursorOrData (GetCursor , PullLogs)	Get the cursor by time; get logs by the cursor and quantity.
log:ListShards	List all available shards in a Logstore.
log:PostLogStoreLogs	Write log data to a specified Logstore.
log:CreateConfig	Create log configuration in a project.
log:UpdateConfig	Update configuration.
log>DeleteConfig	Remove the specified configuration.
log:GetConfig	Obtain configuration details.
log:ListConfig	List all configurations in a project. You can use the parameter to flip pages.
log:CreateMachineGroup	You can create a group of machines to collect logs and deliver configuration.
log:UpdateMachineGroup	Update machine group information.
log>DeleteMachineGroup	Delete a machine group
log:GetMachineGroup	View specific machine group information .
log:ListMachineGroup	View the machine group name list.
log:ListMachines	Get the status information of machines that belong to the user and connect to the server under the machine group.
log:ApplyConfigToGroup	Apply the configuration to the machine group.
log:RemoveConfigFromGroup	Remove configuration from a machine group.
log:GetAppliedMachineGroups	Get the name of the configuration that has been applied on the machine group.
log:GetAppliedConfigs	Obtain the name of the configuration applied to a machine group.
log:GetShipperStatus	Query the LogShipper task status.
log:RetryShipperTask	Rerun failed LogShipper tasks.
log:CreateConsumerGroup	Create a consumer group in a specified Logstore.

Action	Description
log:UpdateConsumerGroup	Modify the attributes of a specified consumer group.
log:DeleteConsumerGroup	Delete a specified consumer group.
log:ListConsumerGroup	Query all consumer groups of a specified Logstore.
log:ConsumerGroupUpdateCheckPoint	Update the checkpoints of a shard of consumer groups under a specified project and Logstore.
log:ConsumerGroupHeartBeat	Send a heartbeat packet to the server for a specified consumer.
log:GetConsumerGroupCheckPoint	Obtain the checkpoints of one or all shards consumed by a specified consumer group.
log>CreateIndex	Create an index for a specified Logstore.
log>DeleteIndex	Create an index for a specified Logstore.
log:GetIndex	Query the index of a specified Logstore.
log:UpdateIndex	Update an index for a specified Logstore.
log>CreateSavedSearch	Create a SavedSearch.
log:UpdateSavedSearch	Update a SavedSearch.
log:GetSavedSearch	View specific SavedSearch information.
log>DeleteSavedSearch	Delete a SavedSearch.
log:ListSavedSearch	List all SavedSearches in a Project.
log>CreateDashboard	Create a Dashboard.
log:UpdateDashboard	Update a Dashboard.
log:GetDashboard	View specific Dashboard information.
log>DeleteDashboard	Delete a Dashboard.
log:ListDashboard	List all Dashboards in a Project.

1.15.4 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 }

Config

Action	Resource
Log: createconfig	acs:log:\${ regionName }:\${ projectOwnerAliUid }:project/\${ projectName }/logtailconfig/*
log:UpdateConfig	acs:log:\${ regionName }:\${ projectOwnerAliUid }:project/\${ projectName }/logtailconfig/\${ logtailConfigName }
log:DeleteConfig	acs:log:\${ regionName }:\${ projectOwnerAliUid }:project/\${ projectName }/logtailconfig/\${ logtailConfigName }
log:GetConfig	acs:log:\${ regionName }:\${ projectOwnerAliUid }:project/\${ projectName }/logtailconfig/\${ logtailConfigName }
log:ListConfig	acs:log:\${ regionName }:\${ projectOwnerAliUid }:project/\${ projectName }/logtailconfig/*

machinegroup

Actions	Resources
log:CreateMachineGroup	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/*
log:UpdateMachineGroup	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/\${ machineGro upName }
log>DeleteMachineGroup	acs:log:\${regionName}:\${projectOwn erAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}
log:GetMachineGroup	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/\${ machineGro upName }
log:ListMachineGroup	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/*
log:ListMachines	acs:log:\${regionName}:\${projectOwn erAliUid}:project/\${projectName}/machinegroup/\${machineGroupName}

Interactive APIs for configuration and machine group

Actions	Resources
log:ApplyConfigToGroup	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/logtailconfig/\${ logtailCon figName } acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/\${ machineGro upName }

Actions	Resources
log:RemoveConfigFromGroup	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/logtailconfig/\${ logtailCon figName } acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/\${ machineGro upName }
log:GetAppliedMachineGroups	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/logtailconfig/\${ logtailCon figName }
log:GetAppliedConfigs	acs:log:\${ regionName }:\${ projectOwn erAliUid }:project/\${ projectNam e }/machinegroup/\${ machineGro upName }

1.16 STS access mode

1.16.1 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 on your resources, and can use all APIs described 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 [#unique_147](#).

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

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 : PostLogSto reLogs ",
      "Resource": "*",
      "Effect": "Allow "
    }
  ]
},
...
```

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

```
...
{
  "Version": " 1 ",
  "Statement": [
    {
      "Action": [
        "log : GetCursorO rData ",
        "log : CreateCons umerGroup ",
        "log : ListConsum erGroup ",
        "log : ConsumerGr oupUpdateC heckPoint ",
        "log : ConsumerGr oupHeartBe at ",
        "log : GetConsume rGroupChec kPoint ",
        "log : UpdateCons umerGroup "
      ]
      "Resource": "*",
      "Effect": "Allow "
    }
  ]
}
```

```
} , ,
```

How to set a resource is described below:

- Configure the resource as follows:
- 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 Limits](#)

Call a Log Service interface

[Log Service SDK Instructions](#)

Code example

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

[Code link](#)

1.17 Common resources

1.17.1 Data model

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

Region

Region is the service node of Alibaba Cloud. By deploying services in different Alibaba Cloud regions, you can get your services closer to the customers, resulting in lower access latency and a better user experience. Currently, Alibaba Cloud provides multiple regions across the country.

Project

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

Logstore

The Logstore is a unit in Log Service to collect, store, and consume logs. Each logstore belongs to one project, and multiple logstores can be created for each project. You can create multiple logstores for one project according to actual needs. The common practice is to create an independent logstore for each type of log in one application. For example, 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

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

- **Topic:** A user-defined field used to mark multiple logs. For example, access logs can be marked according to the sites. This field is a null string by default (the null string 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. Content is composed of one or more content items, and each content item is composed of 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. This field is blank by default.

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

Table 1-6: Log field

Data domain	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.

Data domain	Requirement
source	A UTF-8 encoded string up to 128 bytes.
content	One or more Key-Value Pair. Key is a UTF-8 encoded string of no more than 128 bytes, which contains only letters, underscores, and numbers and cannot begin with a number. The value is a UTF-8 encoded string up to 1024*1024 bytes.

**Note:**

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

**Note:**

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

Various log formats are used in actual application scenarios. For ease of understanding, the following describes how to map an original Nginx access_log to the log data model in Log Service. Assume that the IP address of your Nginx server is 10.10.10.1. An original log of this server is as follows.

```
10 . 1 . 1 . 1 - - [ 01 / Mar / 2012 : 16 : 12 : 07 + 0800 ] " GET
/ Send ? AccessKeyId =< yourAccessKeyId > 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.

Table 1-7: Log Service log data model

Data domain	Item	Description
topic	“”	Use the default value (empty string).

Data domain	Item	Description
time	1330589527	The precise log generation time (in seconds), which is converted from the timestamp of the original log.
source	“10.10.10.1”	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.

Table 1-8: Custom key-value pair

Key	Value
ip	“10.1.1.1”
method	“GET”
status	“200”
length	“5”
ref_url	“_ “
browser	“Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firef

Logs

A collection of multiple logs.

Loggroup

A group of logs.

LogGroupList

A 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.

Table 1-9: Supported encoding method

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

For more information of Protobuf format, see [#unique_68](#).



Note:

Protobuf does not require the key-value pair to be unique. You must avoid such a situation. Otherwise, the behavior is undefined.

Protobuf follows the order of field number to encode fields in one message. Otherwise, data parsing may fail.

1.17.2 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    LogTag
{
    required    string    Key    = 1 ;
    required    string    Value    = 2 ;
}

message    LogGroup
{
    repeated    Log    Logs    = 1 ;
    optional    string    Reserved    = 2 ; //    reserved    fields
    optional    string    Topic    = 3 ;
    optional    string    Source    = 4 ;
    repeated    LogTag    LogTags    = 6 ;
}

message    LogGroupList

```

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

**Note:**

- Protocol Buffer 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](#).

1.17.3 Logstore

The Logstore naming rules are as follows:

- The name contains only 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,
  "autoSplit": true,
  "maxSplitShard": 64,
  "createTime": 1439538649,
  "lastModifyTime": 1439538649
}
```

Parameter definitions

Table 1-10: 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 in days. The minimum value is 1.
shardCount	Integer	Yes	The log data service unit.

Parameter name	Type	Required	Description
autoSplit	Bool	No	Determines whether to automatically split a shard.
maxSplitShard	Int	No	The maximum number of shards for automatic split, which is in the range of 1 to 64. You must specify this parameter when autoSplit is true.
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).

1.17.4 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, 500 times/s
- Read: 10 MB/s, 100 times/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:

Table 1-11: Example of the complete resource:

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

1.17.5 Logtail configuration

You can create up to 100 Logtail configurations for a project. The configuration name must be unique in a project.

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

Logtail configuration naming rules are as follows:

- The name can contain only lowercase letters, digits, hyphens (-), and underscores (_).
- The name must start and end with a lowercase letter or digit.
- The name must be 3 to 63 bytes in length.

A configuration example is as follows:

```
{
  "configName": "logConfigName",
  "outputType": "LogService",
  "inputType": "file",
  "inputDetail": {
    "logPath": "/logPath",
    "filePattern": "access.log",
    "logType": "json_log",
    "topicFormat": "default",
    "discardUnmatched": false,
    "enableRawLog": true,
    "fileEncoding": "gbk",
    "maxDepth": 10
  },
  "outputDetail": {
    "projectName": "test-project",
    "logstoreName": "test-logstore"
  }
}
```

The configuration in JSON mode is used as an example. For more information, see [Configuration examples](#).

Logtail configuration parameters

Parameter	Type	Required	Description
configName	String	Yes	The name of the Logtail configuration, which must be unique in a project.
logSample	String	No	The log sample of the Logtail configuration.

Parameter	Type	Required	Description
inputType	String	Yes	The input type of the Logtail configuration. Valid values: <code>plugin</code> and <code>file</code> .
inputDetail	Object	Yes	The configuration of the input type.
outputType	String	Yes	The output type of the Logtail configuration. Valid value: <code>LogService</code> .
outputDetail	Object	Yes	The configuration of the output type.
createTime (output-only)	Int32	No	The time when the Logtail configuration was created. This parameter is returned by the server and cannot be set.
lastModifyTime (output-only)	Int32	No	The time when the Logtail configuration was last modified. This parameter is returned by the server and cannot be set.

Basic configuration of the inputDetail parameter

The following table describes the basic configuration of the inputDetail parameter for all input types.

Parameter	Type	Required	Description
filterKey	Array	No	The key used to filter logs. A log is collected only when the key value matches the regular expression specified by the corresponding filterRegex parameter.
filterRegex	Array	No	The regular expression corresponding to each filterKey. The length of filterRegex must be the same as that of filterKey.

Parameter	Type	Required	Description
shardHashKey	Array	No	The key that is specified when log data is written in KeyHash mode. Log data is written in LoadBalance mode by default. If you specify this parameter, log data is written in KeyHash mode. Valid values: <code>__topic__</code> , <code>__hostname__</code> , and <code>__source__</code> .
enableRawLog	Boolean	No	Specifies whether to upload the raw log.
sensitive_keys	Array	No	The desensitization configuration, which is a <code>SensitiveKey</code> array. For more information about how to configure <code>SensitiveKey</code> , see the following table.
mergeType	String	No	The aggregation method. Logs are aggregated by topic by default. Valid values: <code>topic</code> and <code>logstore</code> .
delayAlarm Bytes	Integer	No	The alert threshold for slippage on log collection. Default value: <code>209715200</code> , which indicates 200 MB.
adjustTime zone	Boolean	No	Specifies whether to modify the time zone of logs. This parameter is required only when time needs to be parsed.
logTimezone	String	No	The offset of the time zone. If the time zone of logs is UTC+8, set this parameter to <code>GMT + 08 : 00</code> .
priority	Integer	No	The priority for sending logs. Default value: <code>0</code> . If you need to specify a Logtail configuration with a high priority, set this parameter to <code>1</code> .

The following table describes how to configure SensitiveKey.

Parameter	Type	Required	Description
key	String	Yes	The name of the log key.
type	String	Yes	The desensitization method. Valid values: <code>const</code> and <code>md5</code> . If you set this parameter to <code>const</code> , the sensitive content is replaced with the value of the <code>const</code> parameter. If you set this parameter to <code>md5</code> , the sensitive content is replaced with the corresponding MD5 hash value.
regex_begin	String	Yes	The prefix of the sensitive content.
regex_content	String	Yes	The regular expression for the sensitive content.
all	Boolean	Yes	Specifies whether to replace all sensitive content in the field specified by the key parameter. We recommend that you set this parameter to <code>true</code> .
const	String	No	The content used to replace the sensitive content. This parameter is required if you set the <code>type</code> parameter to <code>const</code> .

For example, a log contains the `content` field, whose value is `[{' account ':' 1812213231 432969 ', ' password ':' 04a23f38 '}, {' account ':' 1812213685 634 ', ' password ':' 123a '}]`. To desensitize `password`, set `SensitiveKey` as follows:

```
" key " : " content "
" type " : " const "
" regex_begin " : "' password ':'"
" regex_content " : "[^']*"
" all " : true
```

```
" const " : "*****"
```

The log content after desensitization is as follows:

```
[{' account ':' 1812213231 432969 ',' password ':'*****'}, {'  
 account ':' 1812213685 634 ',' password ':'*****'}]
```

Configuration of the inputDetail parameter for text files

The following table describes the basic configuration of the inputDetail parameter for text files.

Parameter	Type	Required	Description
logType	String	Yes	The log collection mode. This parameter is required if you set the inputType parameter to file. Valid values: <code>json_log</code> , <code>apsara_log</code> , <code>common_reg_log</code> , and <code>delimiter_log</code> .
logPath	String	Yes	The parent directory of logs to be collected, for example, <code>/var / logs /</code> .
filePattern	String	Yes	The pattern of the log file, for example, <code>access *. log</code> .
topicFormat	String	Yes	The method for generating the topic of logs. Valid values: <code>none</code> : indicates that the topic is set to a null string. <code>default</code> : indicates that the log path is used as the topic. <code>group_topi c</code> : indicates that the topic attribute of the machine group that applies the Logtail configuration is used as the topic. Another value: indicates that a part of the log path is used as the topic, for example, <code>/var / log /(.*). log</code> .
timeFormat	String	No	The time format of logs, for example, <code>% Y /% m /% d % H :% M :% S</code> .

Parameter	Type	Required	Description
preserve	Boolean	No	The timeout period of the monitored directory. A value of true indicates that the monitored directory never times out. A value of false indicates that the monitored directory is regarded as timing out if it is not updated within 30 minutes. Default value: true.
preserveDepth	Integer	No	The depth of the monitored directory if you set the preserve parameter to true. The maximum value is 3.
fileEncoding	String	No	The file encoding method. Valid values: <code>utf8</code> and <code>gbk</code> .
discardUnmatch	Boolean	No	Specifies whether to discard logs that fail to match the specified filter conditions.
maxDepth	Integer	No	The maximum depth of the monitored directory. Valid values: [0, 1000]. A value of 0 indicates that only the current directory is monitored.
delaySkipBytes	Integer	No	The discarding threshold for slippage on log collection. Default value: 0, which indicates that lagged data is not discarded. If the amount of lagged data exceeds the value of this parameter, the lagged data is discarded.
isDockerFile	Boolean	No	Specifies whether the log file is in a container. Default value: false. For more information, see Container text logs .

Parameter	Type	Required	Description
dockerIncludeLabel	Object	No	The whitelist of container labels . Logtail collects logs from Docker containers whose labels are in the whitelist. If you do not specify this parameter, Logtail collects logs from all containers.
dockerExcludeLabel	Object	No	The blacklist of container labels . Logtail does not collect logs from Docker containers whose labels are in the blacklist. If you do not specify this parameter , Logtail collects logs from all containers.
dockerIncludeEnv	Object	No	The whitelist of container environment variables. Logtail collects logs from Docker containers whose environment variables are in the whitelist . If you do not specify this parameter, Logtail collects logs from all containers.
dockerExcludeEnv	Object	No	The blacklist of container environment variables. Logtail does not collect logs from Docker containers whose environment variables are in the blacklist. If you do not specify this parameter, Logtail collects logs from all containers.

- Configuration in full regex mode or simple mode

The following table describes the specific configuration in full regex mode or simple mode.

Parameter	Type	Required	Description
key	Array	Yes	The keys in the log extraction results.
logBeginRegex	String	No	The regular expression used to specify the starting header of a cross-line log.

Parameter	Type	Required	Description
regex	String	No	The regular expression used to extract log fields.

The `simple mode` is a special case of the full regex mode. The following example shows the configuration in `simple mode` :

```
" key " : [" content "]  
" logBeginRe_gex " : ".*"  
" regex " : "(. *)"
```

- Configuration in JSON mode

The following table describes the specific configuration in JSON mode.

Parameter	Type	Required	Description
timeKey	String	No	The key that specifies the time field.

- Configuration in delimiter mode

The following table describes the specific configuration in delimiter mode.

Parameter	Type	Required	Description
separator	String	No	The delimiter used to delimit the fields of each log.
quote	String	Yes	The quote used to enclose log fields that contain the delimiter .
key	Array	Yes	The list of keys in the log extraction results.
timeKey	String	Yes	The key that specifies the time field, which must be in the key list.
autoExtend	Boolean	No	Specifies whether to automatically increase the number of keys if the actual number of parsed log fields is greater than the number of configured keys.

- Configuration in Apsara mode

The following table describes the specific configuration in Apsara mode.

Parameter	Type	Required	Description
logBeginRegex	String	No	The regular expression used to specify the starting header of a cross-line log.

Configuration of the outputDetail parameter

The following table describes how to configure the project and Logstore for output logs.

Parameter	Type	Required	Description
projectName	String	Yes	The name of the project, which must be the name as the requested project. Otherwise, an error is returned.
logstoreName	String	Yes	The name of the Logstore.

Configuration examples

- JSON mode:

```
{
  " configName ": " logConfigName ",
  " outputType ": " LogService ",
  " inputType ": " file ",
  " inputDetail ": {
    " logPath ": "/ logPath ",
    " filePattern ": " access . log ",
    " logType ": " json_log ",
    " topicFormat ": " default ",
    " discardUnmatched ": false ,
    " enableRawLog ": true ,
    " fileEncoding ": " gbk ",
    " maxDepth ": 10
  },
  " outputDetail ": {
    " projectName ": " test - project ",
    " logstoreName ": " test - logstore "
  }
}
```

- Delimiter mode:

```
{
  " configName ": " logConfigName ",
  " logSample ": " testlog ",
  " inputType ": " file ",
  " outputType ": " LogService ",
```

```

    " inputDetail ": {
      " logPath ": "/ logPath ",
      " filePattern ": "*",
      " logType ": " delimiter_log ",
      " topicFormat ": " default ",
      " discardUnmatched ": true ,
      " enableRawLog ": true ,
      " fileEncoding ": " utf8 ",
      " maxDepth ": 999 ,
      " separator ": ",",
      " quote ": "\"",
      " key ": [
        " test ",
        " test2 "
      ],
      " autoExtend ": true
    },
    " outputDetail ": {
      " projectName ": " test - project ",
      " logstoreName ": " test - logstore "
    }
  }
}

```

• Full regex mode:

```

{
  " configName ": " logConfigName ",
  " outputType ": " LogService ",
  " inputType ": " file ",
  " inputDetail ": {
    " logPath ": "/ logPath ",
    " filePattern ": "*",
    " logType ": " common_regex_log ",
    " topicFormat ": " default ",
    " discardUnmatched ": false ,
    " enableRawLog ": true ,
    " fileEncoding ": " utf8 ",
    " maxDepth ": 10 ,
    " key ": [
      " content "
    ],
    " logBeginRegex ": ".*",
    " regex ": "(.*)"
  },
  " outputDetail ": {
    " projectName ": " test - project ",
    " logstoreName ": " test - logstore "
  }
}

```

• Plug-in mode (Docker standard output):

```

{
  " configName ": " logConfigName ",
  " outputType ": " LogService ",
  " inputType ": " plugin ",
  " inputDetail ": {
    " plugin ": {
      " inputs ": [
        {
          " detail ": {
            " ExcludeEnv ": null ,
            " ExcludeLabel ": null ,

```

```

        " IncludeEnv ": null ,
        " IncludeLab el ": null ,
        " Stderr ": true ,
        " Stdout ": true
    },
    " type ": " service_do cker_stdou t "
}
]
}
},
" outputData il ": {
    " projectNam e ": " test - project ",
    " logstoreNa me ": " test - logstore "
}
}

```

1.17.6 Machine group

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 method to understand, but the IP address may be duplicated in the VPC and other environments.
- **UUID (machine-uniqueid):** The UUID in DMI devices. For 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 ",
    " userdefine d - id " : " testuserde finedid1 ",
    " lastHeartb eatTime ": 1397781420
}

```

The machine property parameters are as follows:

Table 1-12: Parameter description

Parameter	Type	Description
ip	string	The IP address corresponding to the machine hostname.

Parameter	Type	Description
uuid	string	The unique primary key marked by the machine, uploaded by Logtail.
userdefined-id	string	The user-defined machine identification, uploaded by Logtail.
lastHeartbeatTime(output-only)	integer	The last heartbeat time of the machine (the number of seconds since the epoch time).

machinegroup

Machinegroup is the machine group that belongs to the current user in the project. 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.

Machinegroup 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" : "testgroup topic"
  },
  "machineIdentifyType" : "ip",
  "machineList" : [
    "ip1",
    "ip2",
    ...
  ],
  "createTime" : 1431705075,
  "lastModifyTime" : 1431705075
}
```

```
}

```

Table 1-13: Description

Attribute name	Type	Required	Description
groupName	string	Yes	The machine group name, which must be unique in the 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)	int	No	The resource creation time.
lastModifyTime(output-only)	int	No	The resource server update time.

groupAttribute is described as follows.

Table 1-14: 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 on, which is empty by default.