Alibaba Cloud Log Service

SDK Reference

Issue: 20190321

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.
`Courier font`	It is used for commands.	Run the `cd / d C :/ windows` command to enter the Windows system folder.
Italics	It is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *[-all|-t]*

Style	Description	Example
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich {stand | slave}`

Contents

1 Basic Descriptions

1.1 Overview

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

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

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

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

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

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

- Unifiederror handling method , allowing you to handle request exceptions in the method that languages are familiar with.
- Currently, SDKs in all languages only support synchronous requests.

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

SDK language	Relevant document	Source code
Java	Java SDK, UserGuide	GitHub
.NET	.NET SDK, UserGuide	GitHub
PHP	PHP SDK	GitHub
Node. js		GitHub
Python	Python SDK, UserGuide	GitHub
C	Usage instructions	GitHub
GO	Usage instructions	GitHub
iOS	iOS SDK	GitHub
Android		GitHub

1.2 Configurations

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

include the following items:

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

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

Service endpoint

When using SDKs, you must identify the region where the Log Service project to be accessed resides (such as China East 1 (Hangzhou) or China North 1 (Qingdao)) and then select the Log Service endpoint that matches with the region to initialize

the client. the client. The service endpoint is defined in the same way as the service endpoint of APIs *Service endpoint* .

- When selecting an endpoint for the client, make sure that the region where the project to be accessed resides is the same as the region that corresponds to the endpoint. Otherwise, you cannot use SDK to access your specified project.
- The client can only specify the service endpoint when being built, you must use different endpoints to build different clients if you want to access projects in different regions.
- Currently, all the API service endpoints only support HTTP.
- You can also use an intranet endpoint to avoid Internet bandwidth overhead if you are using SDKs in an Alibaba Cloud Elastic Compute Service (ECS) instance. For more information, see *Service endpoint*.

AccessKey

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

Note:

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

Example

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

```
AccessKeyI  d  = " bq2sjzesjm  o **************"
```

```
 AccessKeyS  ecret  = " 4fdO2fTDDn  ZPU /*************"
```

The corresponding client instance can be instanced as follows:

Java:

```
 String  endpoint  = " cn - hangzhou . log . aliyuncs . com "; //
 The  Log  Service  endpoint  of  region  China  East  1  (
 Hangzhou ).
 String  accessKeyI  d  = " bq2sjzesjm  o86kq35beh  upbq "; // Your
 AccessKey  ID .
 String  accessKeyS  ecret  = " 4fdO2fTDDn  ZPU / L7CHNdemB2  Nsk
 =";// Your  AccessKey  Secret .
 Client  client  =  new  Client ( endpoint ,  accessKeyI  d ,
 acccessKey  Secret );
// Use  client  to  operate  Log  Service  project ...
```

NET(C#):

```
 String  endpoint  = " cn - hangzhou . log . aliyuncs . com "; //
 The  Log  Service  endpoint  of  region  China  East  1  (
 Hangzhou ).
 String  accessKeyI  d  = " bq2sjzesjm  o86kq35beh  upbq "; // Your
 AccessKey  ID .
 String  accessKeyS  ecret  = " 4fdO2fTDDn  ZPU / L7CHNdemB2  Nsk
 =";// Your  AccessKey  Secret .
 SLSClient  client  =  new  SLSClient ( endpoint ,  accessKeyI  d ,
 accessKeyS  ecret );
// use  client  to  operate  sls  project ......
```

PHP:

```
$ endpoint  = ' cn - hangzhou . log . aliyuncs . com '; // The  Log
 Service  endpoint  of  region  China  East  1  ( Hangzhou ).
$ accessKeyI  d  = ' bq2sjzesjm  o86kq35beh  upbq '; // Your
 AccessKey  ID .
$ accessKey  = ' 4fdO2fTDDn  ZPU / L7CHNdemB2  Nsk =';// Your
 AccessKey  Secret .
$ client  =  new  Aliyun_Sls  _Client ($ endpoint , $ accessKeyI  d ,
 $ accessKey );
// use  client  to  operate  sls  project ......
```

Python:

```
// Use  client  to  operate  Log  Service  project ...
 endpoint  = ' regionid . example . com '
#  The  Log  Service  endpoint  of  region  China  East  1  (
 Hangzhou ).
 accessKeyI  d  = ' bq2sjzesjm  o *************'
#  Your  AccessKey  ID .
 accessKey  = ' 4fdO2fTDDn  ZPU /*************'
 client  =  LogClient ( endpoint ,  accessKeyI  d ,  accessKey )
```

```
# use   client   to   operate   log   project ......
```

1.3 Handle errors

Possible SDK errors are classified as follows:

- Errors returned by the Log Service. This type of errors is returned by the Log Service and handled by SDKs. For more information about this error type, see the *Common error codes* of the Log Service APIs and the descriptions of each API.
- Network errors that occur when SDKs send requests to the Log Service. This type of errors includes network interruptions and Log Service return timeout.
- Errors that are produced by SDKs and related to platforms or languages, for example, memory overflow.

Currently, the SDKs in various languages handle errors by throwing exceptions. The specific principles are as follows:

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

LogException

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

- Error code: Indicates the error type. For errors returned by Log Service, the error code is the same as that returned by APIs. For network errors of SDK requests, the error code is "Requesterror ". For more information, see the complete API reference of each language.
- Error message: Indicates the message that comes with an error. For errors returned by Log Service, the error message is the same as that returned by APIs. For network errors of SDK requests, the error message is “request is failed”. For more information, see the complete API reference of each language.
- Request ID: Indicates the request ID in Log Service that corresponds to the current error. This ID is valid only when Log Service returns an error message. Otherwise,

it is an empty string. When a request error occurs, you can provide the request ID to the Log Service team to troubleshoot the problem.

Request failure and retry

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

Example

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

Java:

```
// Other code ...
String accessId = " your_acces s_id "; // TODO : Use your
Alibaba Cloud AccessKey ID .
String accessKey = " your_acces s_key "; // TODO : Use your
Alibaba Cloud AccessKey Secret .
String project = " big - game ";
String endpoint = " cn - hangzhou . sls . aliyuncs . com ";
int max_retrie s = 3 ;
/*
* Construct a client
*/
Client client = new client ( adord , accesskey , endpoint
);
ListLogSto resRequest lsRequest = new ListLogSto resRequest (
project );
for ( int i = 0 ; i < max_retrie s ; i ++)
{
    try
    {
        ListLogSto resRespons e res = client . ListLogSto res
( lsRequest )
       // TODO : Process the returned response ...
        break ;
    }
    catch ( LogExcepti on ex )
    {
        if ( e . GetErrorCo de () == " RequestErr or ")
       {
            if ( i == max_retrie s - 1 )
           {
                System . out . println (" request is still
failed after all retries .");
                break ;
           }
            else
```

```
                System . out . println (" request   error   happens
 ,  retry   it !") ;
        }
         else
        {
             System . out . println (" error   code  :" +  e .
 GetErrorCo  de ());
             System . out . println (" error   message  :" +  e .
 GetErrorMe  ssage ());
             System . out . println (" error   requestId  :" +  e .
 GetRequest  Id ());
             break ;
        }
    }
     catch (...)
    {
         System . out . println (" unrecovera  ble   exception   when
   listing   logstores .");
         break ;
    }
}
// Other   code ...
```

. NET(C#):

```
// Other   code ...
 String   accessId  = " your_acces  s_id "; // TODO :  Use   your
 Alibaba   Cloud   AccessKey   ID .
 String   accessKey  = " your_acces  s_key "; // TODO :  Use   your
 Alibaba   Cloud   AccessKey   Secret .
 String   project  = " big - game ";
 String   endpoint  = " cn - hangzhou . sls . aliyuncs . com ";
 int   max_retrie  s  =  3 ;
// Construct   a   client
 SLSClient   client  =  new   SLSClient ( endpoint ,  accessId ,
 accessKey );
 ListLogsto  resRequest   request  =  new   ListLogsto  resRequest ();
 request . Project  =  project ;
 for  ( int   i  =  0 ;  i  <  max_retrie  s ;  i ++)
{
     try
    {
         ListLogsto  resRespons  e   response  =  client . ListLogsto
 res ( request );
        // TODO :  Process   the   returned   response ...
         break ;
    }
     catch ( LogExcepti  on   ex )
    {
         if  ( e . errorCode  == " SLSRequest  Error ")
        {
             if  (  i  ==  max_retrie  s  -  1 )
            {
                 Console . Writeline (“ request   is   still   failed
   after   all   retries .”);
                 break ;
            }
             else
            {
                 Console . Writeline (" request   error   happens ,
 retry   it !") ;
            }
        }
```

```
         else
        {
             Console . Writeline (" error   code  :" +  e . errorCode
 ;
             Console . Writeline (" error   message  :" +  e . Message
 ;
             Console . Writeline (" error   requestId  :" +  e .
 RequestId ;
             break ;
        }
    }
     catch (...)
    {
         Console . Writeline (" unrecovera  ble   exception   when
 listing   logstores .");
        break ;
    }
}
// Other   code ...
```

PHP:

```
<?  php
// Other   code ...
$ endpoint  = ' cn - hangzhou . sls . aliyuncs . com ';
$ accessId  = ' your_acces  s_id '; //  TODO :  Use   your   Alibaba
 Cloud   AccessKey   ID .
$ accessKey  = ' your_acces  s_key '; // TODO :  Use   your   Alibaba
   Cloud   AccessKey   Secret .
$ maxRetries  =  3 ;
//  Build   a   client .
$ client  =  new   Aliyun_Sls  _Client ($ endpoint , $ accessId , $
 accessKey );
$ project  = ' big - game ';
$ request  =  new   Aliyun_Sls  _Models_Li  stLogstore  sRequest ($
 project );
 for ($ i  =  0 ; $ i  < $ maxRetries ; ++$ i )
{
     try
    {
        $ response  = $ client -> ListLogsto  res ($ request );
        // TODO :  Process   the   returned   response ...
         break ;
    }
     catch  ( Aliyun_Sls  _Exception  $ e )
    {
         if  ($ e -> getErrorCo  de ()==' RequestErr  or ')
        {
             if  ($ i + 1  == $ maxRetries )
            {
                 echo  " error   code  :" . $ e -> getErrorCo  de
 () .  PHP_EOL ;
                 echo  " error   message  :" . $ e -> getErrorMe
 ssage () .  PHP_EOL ;
                 break ;
            }
             echo  ' request   error   happens ,  retry   it !' .
 PHP_EOL ;
        }
         else
        {
             echo  " error   code  :" . $ e -> getErrorCo  de () .
 PHP_EOL ;
```

```
            echo  " error   message  :" . $ e -> getErrorMe  ssage
 () .  PHP_EOL ;
            echo  " error   requestId  :" . $ e -> getRequest  Id
 () .  PHP_EOL ;
            break ;
         }
    }
     catch  ( Exception  $ ex )
    {
         echo  ' unrecovera  ble   exception   when   listing
 logstores .' .  PHP_EOL ;
         var_dump ($ ex );
         break ;
    }
}
// Other   code ...
```

Python:

```
// Other   code ...
 endpoint  = ' cn - hangzhou . sls . aliyuncs . com '
 accessId  = ' your_acces  s_id ' #  TODO :  Use   your   Alibaba
 Cloud   AccessKey   ID .
$ accessKey  = ' your_acces  s_key '; // TODO :  Use   your   Alibaba
   Cloud   AccessKey   Secret .
 maxRetries  =  3
#  Construct   a   client
 client  =  Client ( endpoint ,  accessId ,  accessKey )
 project  = ' big - game '
 lsRequest  =  ListLogsto  resRequest ( project )
 for   i   in   xrange ( maxRetries ):
     try :
         res  =  client . ListLogsto  res ( lsRequest )
        #  TODO :  Process   the   returned   response ...
         break
     except   LogExcepti  on   as   e :
         if   e . getErrorCo  de () == " RequestErr  or ":
             if   i + 1  ==  maxRetries :
                 print  " error   code  :" +  e . getErrorCo  de ()
                 print  " error   message  :" +  e . getErrorMe  ssage
 ()
                 break
             else :
                 print  " request   error   happens ,  retry   it !"
         else :
             print  " error   code  :" +  e . getErrorCo  de ()
             print  " error   message  :" +  e . getErrorMe  ssage ()
             print  " error   requestId  :" +  e . getRequest  Id ()
             break
     except   Exception   as   e :
         print  ' unrecovera  ble   exception   when   listing
 logstores .'
         break
// Other   code ...
```

1.4 Interface regulations

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

1. Build a request by using request parameters.
2. Call the corresponding interface in the SDK and pass in the request in the preceding step.
3. Encapsulate the results returned by the SDK interface into a response and then return the response to the user.

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

Java

```
// Other code ......
String accessId = " your_acces s_id "; // TODO : Use your
Alibaba Cloud AccessKey ID .
String accessKey = " your_acces s_key "; // TODO : Use your
Alibaba Cloud AccessKey Secret .
String project = " your_proje ct "; // TODO : Use your
project name .
String endpoint = " region_end point ";// TODO : Use the
endpoint that correspond s to the region where your
project resides .
// Build a client .
Client client = new Client ( endpoint , accessId , accessKey
);
// Use the request parameter " project " to initialize the
ListLogsto res request class .
ListLogSto resRequest lsRequest = new ListLogSto resRequest (
project );
// Use the request to call the ListLogsto res interface
. The return parameter is the correspond ing response .
ListLogSto resRespons e res = client . ListLogSto res (
lsRequest );
// Access the response to retrieve the request results .
ArrayList < String > names = res . GetLogStor es ();
// Other code ...
```

. NET(C#)

```
// Other code ...
String accessId = " your_acces s_id "; // TODO : Use your
Alibaba Cloud AccessKey ID .
String accessKey = " your_acces s_key "; // TODO : Use your
Alibaba Cloud AccessKey Secret .
String project = " your_proje ct "; // TODO : Use your
project name .
String endpoint = " region_end point ";// TODO : Use the
endpoint that correspond s to the region where your
project resides .
// Construct a client instance .
SLSClient client = new SLSClient ( endpoint , accessId ,
accessKey );
// Use the request parameter " project " to initialize a
ListLogsto res request class .
ListLogSto resRequest lsRequest = new ListLogSto resRequest
();
lsRequest . Project = project ;
```

```
// Use the request instance to call the ListLogsto res
   interface . The return parameter is the correspond ing
   response instance .
 ListLogSto resRespons e res = client . ListLogSto res (
 lsRequest );
// Access the response instance to retrieve the request
   results
 List < String > names = res . Logstores ;
//  Other code ...
```

PHP

```
//  Other code ...
 accessId = " your_acces s_id "; // TODO : Use your Alibaba
 Cloud AccessKey ID .
$ accessKey = " your_acces s_key "; // TODO : Use your Alibaba
   Cloud AccessKey Secret .
$ project = " your_proje ct "; // TODO : Use your project
 name .
$ endpoint = " region_end point ";// TODO : Use the endpoint
 that correspond s to the region where your project
 resides .
// Construct a Log Service client instance .
$ client = new Aliyun_Sls _Client ($ endpoint , $ accessId , $
 accessKey );
// Use the request parameter " project " to initialize a
 ListLogsto res request class .
$ request = new Aliyun_Sls _Models_Li stLogstore sRequest ($
 project );
// Use the request instance to call the ListLogsto res
   interface . The return parameter is the correspond ing
   response instance .
$ response = $ client -> listLogsto res ($ request );
// Access the response instance to retrieve the request
   results
$ names = $ response -> getLogstor es ();
//  Other code ......
```

Python

```
//  Other code ......
 accessId = ' your_acces s_id '; // TODO : Use your Alibaba
 Cloud AccessKey ID .
$ accessKey = " your_acces s_key "; // TODO : Use your Alibaba
   Cloud AccessKey Secret .
 project = ' your_proje ct '; // TODO : Use your project name
 .
 endpoint = ' region_end point ';// TODO : Use the endpoint
 that correspond s to the region where your project
 resides .
#  Build a client .
 client = LogClient ( endpoint , accessId , accessKey )
#  Use the request parameter " project " to initialize the
   ListLogsto res request class .
 lsRequest = ListLogsto resRequest ( project )
#  Use the request to call the ListLogsto res interface
 . The return parameter is the correspond ing response .
 res = client . list_logst ores ( lsRequest )
#  Access the response to retrieve the request results .
 names = res . get_logsto res ();
```

```
//  Other   code ......
```

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

2 Java SDK

Download address

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

Procedure

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

Step 1. Create an Alibaba Cloud account

For more information, see*Sign up with Alibaba Cloud* .

Step 2. Obtain an Alibaba Cloud AccessKey

Before using Log Service Java SDK, you must apply for an Access Key.

Log on to the *Access Key Management page* . Select an AccessKey for for SDK. If you do not have any, create one and make sure the AccessKey is enabled. The AccessKey is used in the following steps and must be kept confidential. For more information about how to use the AccessKey in SDK, see *Preparation* SDK configuration.

This access key will be used in the following steps. It must be kept confidential. See *Configurations* for more information about how to use the AccessKey in SDK.

Step 3. Create a Log Service project and a Logstore

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

For how to create a project and Logstore, see *Preparation*.

Note:

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

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

Step 4. Install the Java development environment

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

Step 5. Install Log Service Java SDK

Install the Log Service Java SDK after you build the Java development environment. Currently, two installation methods are available.

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

```
< dependency >
        < groupId > com . google . protobuf </ groupId >
        < artifactId > protobuf - java </ artifactId >
        < version > 2 . 5 . 0 </ version >
</ dependency >
< dependency >
< groupId > com . aliyun . openservic  es </ groupId >
< artifactId > aliyun - log </ artifactId >
< version > 0 . 6 . 7 </ version >
< exclusions >
       < exclusion >
           < groupId > com . google . protobuf </ groupId >
           < artifactId > protobuf - java </ artifactId >
      </ exclusion >
</ exclusions >
</ dependency >
```

2. You can download the Java SDK package and then directly reference the local package in your Java project.
   a. Click *here* to clone the Java SDK package. Version updates are provided periodically. Use Maven to obtain the latest version.
   b. Extract the downloaded package to a specified directory. The Java SDK does not require installation.
   c. Add all .jar packages (including third-party dependent packages) in the SDK package to your Java project. For detailed instructions, see the corresponding IDE document.

Step 6. Start a new Java project

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

```
package sdksample ;
import java . util . ArrayList ;
import java . util . List ;
import java . util . Vector ;
import java . util . Date ;
import com . aliyun . openservic es . log . Client ;
import com . aliyun . openservic es . log . common .*;
import com . aliyun . openservic es . log . exception .*;
import com . aliyun . openservic es . log . request .*;
import com . aliyun . openservic es . log . response .*;
import com . aliyun . openservic es . log . common . LogGroupDa
ta ;
import com . aliyun . openservic es . log . common . LogItem ;
import com . aliyun . openservic es . log . common . Logs . Log ;
import com . aliyun . openservic es . log . common . Logs . Log .
Content ;
import com . aliyun . openservic es . log . common . Logs .
LogGroup ;
import com . aliyun . openservic es . log . common . Consts .
CursorMode ;
public class sdksample {
    public static void main ( String args []) throws
LogExcepti on , Interrupte dException {
        String endpoint = "< log_servic e_endpoint >"; // Select
  the endpoint that matches the region of the project
  created in the preceding step .
       // Endpoint
        String accessKeyI d = "< your_acces s_key_id >"; // Use
  your Alibaba Cloud AccessKey ID .
        String accessKeyS ecret = "< your_acces s_key_secr et
>"; // Use your Alibaba Cloud AccessKey Secret .
       // AccessKeyS ecret
        String project = "< project_na me >"; // The name of
  the project created in the preceding step .
        String logstore = ""< logstore_n ame >"; // The name
of the Logstore created in the preceding steps .
       // Build a client instance .
        Client client = new Client ( endpoint , accessKeyI d
, accessKeyS ecret );
       // List the names of all LogStores under the
current project
        int offset = 0 ;
        int size = 100 ;
        String logStoreSu bName = "";
        ListLogSto resRequest req1 = new ListLogSto
resRequest ( project , offset , size , logStoreSu bName );
        ArrayList < String > logStores = client . ListLogSto res
( req1 ). GetLogStor es ();
        System . out . println (" ListLogs :" + logStores . toString
() + "\ n ");
       // Write logs
        String topic = "";
        String source = "";
       // Send 10 packages consecutiv ely , with each
package containing 10 logs
```

```
        for ( int i = 0 ; i < 10 ; i ++) {
            Vector < LogItem > logGroup = new Vector < LogItem
>();
            for ( int j = 0 ; j < 10 ; j ++) {
                LogItem logItem = new LogItem (( int ) ( new
Date (). getTime () / 1000 ));
                logItem . PushBack (" index "+ String . valueOf ( j
), String . valueOf ( i * 10 + j ));
                logGroup . add ( logItem );
            }
            PutLogsReq uest req2 = new PutLogsReq uest (
project , logstore , topic , source , logGroup );
            client . PutLogs ( req2 );
            /*
             * You can specify the shard to which data
  is sent by setting the shard HashKey .
              * Data is written to the shard whose
range includes the HashKey . For more informatio n
about the API , see the following interface : public
PutLogsRes ponse
             * PutLogs ( String project , String logStore ,
String topic ,
             * List < logitem > logitems , string source ,
string shardhash //
             * Write data to the shard based on the
hashkey , which may be MD5 ( ip ) or MD5 ( id ).) throws
             * LogExcepti on ;
             */
       }
       // Read the data written to Shard 0 during
the past minute .
        int shard_id = 0 ;
        long curTimeInS ec = System . currentTim eMillis () /
1000 ;
        GetCursorR esponse cursorRes = client . GetCursor (
project , logstore , shard_id , curTimeInS ec - 60 );
        String beginCurso r = cursorRes . GetCursor ();
        cursorRes = client . GetCursor ( project , logstore ,
shard_id , CursorMode . END );
        String endCursor = cursorRes . GetCursor ();
        String curCursor = beginCurso r ;
        while ( curCursor . equals ( endCursor ) == false ) {
            int loggroup_c ount = 2 ; // Read two log
groups at a time .
            BatchGetLo gResponse logDataRes = client .
BatchGetLo g ( project , logstore , shard_id , loggroup_c ount
, curCursor ,
                    endCursor );
            // Read the log group list .
            List < LogGroupDa ta > logGroups = logDataRes .
GetLogGrou ps ();
            for ( LogGroupDa ta logGroup : logGroups ){
                FastLogGro up flg = logGroup . GetFastLog Group
();
                System . out . println ( String . format ("\
tcategory \ t :\ t % s \ n \ tsource \ t :\ t % s \ n \ ttopic \ t :\
t % s \ n \ tmachineUU ID \ t :\ t % s ",
                    flg . getCategor y (), flg . getSource (),
flg . getTopic (), flg . getMachine UUID ()));
                System . out . println (" Tags ");
                for ( int tagIdx = 0 ; tagIdx < flg .
getLogTags Count (); ++ tagIdx ) {
                    FastLogTag logtag = flg . getLogTags ( tagIdx
);
```

```
                    System . out . println ( String . format ("\ t %
s \ t :\ t % s ",  logtag . getKey (),  logtag . getValue ()));
                }
                 for  ( int   lIdx  =  0 ;  lIdx  <  flg . getLogsCou
nt (); ++ lIdx ) {
                    FastLog   log  =  flg . getLogs ( lIdx );
                    System . out . println ("--------\ nLog : " +
lIdx  + ",  time : " +  log . getTime () + ",  GetContent  Count : "
+  log . getContent  sCount ());
                    for  ( int   cIdx  =  0 ;  cIdx  <  log .
getContent  sCount (); ++ cIdx ) {
                        FastLogCon  tent   content  =  log .
getContent  s ( cIdx );
                        System . out . println ( content . getKey ()
+ "\ t :\ t " +  content . getValue ());
                    }
                }
            }
             String   next_curso  r  =  logDataRes . GetNextCur  sor
();
             System . out . println (" The   Next   cursor :" +
next_curso  r );
             curCursor  =  next_curso  r ;
        }
        // ! ! !  //  Note :  You   can   call   the   following
interface   only   after   the   index   function   is   enabled .
        // Wait   1   minute   until   logs   are   queryable
         try  {
             Thread . sleep ( 60  *  1000 );
        }  catch  ( Interrupte  dException   e ) {
             e . printStack  Trace ();
        }
        // Query   log   distributi  on
         String   query  = "< The   query   keyword .  To   query
all   the   contents ,  use   an   empty   string   here .>";
          int   from  = ( int ) ( new   Date (). getTime () /  1000  -
300 );
         int   to  = ( int ) ( new   Date (). getTime () /  1000 );
         GetHistogr  amsRespons  e   res3  =  null ;
         while  ( true ) {
             GetHistogr  amsRequest   req3  =  new   GetHistogr
amsRequest ( project ,  logstore ,  topic ,  query ,  from ,  to );
             res3  =  client . GetHistogr  ams ( req3 );
             if  ( res3  ! =  null  &&  res3 .  IsComplete  d ()) //
IsComplete  d ()
            //  If   IsComplete  d ()  returns  " true ",  the   query
  results   are   accurate .
            // If  " false "  is   returned ,  query   the   results
again .
            {
                 break ;
            }
             Thread . sleep ( 200 );
        }
         System . out . println (" Total   count   of   logs   is  "
+  res3 .  GetTotalCo  unt ());
         for  ( Histogram   ht  :  res3 .  GetHistogr  ams ()) {
             System . out . printf (" from  % d ,  to  % d ,  count  %
d .\ n ",  ht . GetFrom (),  ht . GetTo (),  ht . GetCount ());
        }
        // Query   log   data
         long   total_log_  lines  =  res3 .  GetTotalCo  unt ();
         int   log_offset  =  0 ;
```

```
        int log_line = 10 ; // log_line the maximum value
  is 100 and 100 rows of data are obtained each
time . If you want to read more data , use offset
to flip the page . Offset and lines are only valid
  for keyword queries , and if SQL queries are used ,
they are not valid . To return more data in a SQL
  query , use the limit syntax .
        while ( log_offset <= total_log_ lines ) {
            GetLogsRes ponse res4 = null ;
           // Read 10 lines of logs at a time for
each log offset . If the read operation fails , it is
  retried three times at most .
            for ( int retry_time = 0 ; retry_time < 3 ;
retry_time ++) {
                GetLogsReq uest req4 = new GetLogsReq uest (
project , logstore , from , to , topic , query , log_offset ,
                        log_line , false );
                res4 = client . GetLogs ( req4 );
                if ( res4 ! = null && res4 . IsComplete d ())
{
                    break ;
                }
                Thread . sleep ( 200 );
            }
            System . out . println (" Read log count :" + String
. valueOf ( res4 . GetCount ()));
            log_offset += log_line ;
       }
       // Enable the analysis function . You can use
the SQL function only after enabling the analysis
function . You can enable the analysis function in
the console or by using SDKs . // Use the analysis
function .
        IndexKeys indexKeys = new IndexKeys ();
        ArrayList < String > tokens = new ArrayList < String
>();
        tokens . add (",");
        tokens . add (".");
        tokens . add ("#");
        IndexKey keyContent = new IndexKey ( tokens , false ,"
text ");
        indexKeys . AddKey (" index0 ", keyContent );
        keyContent = new IndexKey ( new ArrayList < String >(),
false ," long ");
        indexKeys . AddKey (" index1 ", keyContent );
        keyContent = new IndexKey ( new ArrayList < String >(),
false ," double ");
        indexKeys . AddKey (" index2 ", keyContent );
        IndexLine indexLine = new IndexLine ( new ArrayList <
String >(), false );
        Index index = new Index ( 7 , indexKeys , indexLine );
        CreateInde xRequest createInde xRequest = new
CreateInde xRequest ( project , logstore , index );
        client . CreateInde x ( createInde xRequest );
        // Use the analysis function .
        GetLogsReq uest req4 = new GetLogsReq uest ( project
, logstore , from , to , "", " index0 : value | select avg (
index1 ) as v1 , sum ( index2 ) as v2 , index0 group by
index0 ");
        GetLogsRes ponse res4 = client . GetLogs ( req4 );
        if ( res4 ! = null && res4 . IsComplete d ()){
            for ( QueriedLog log : res4 . GetLogs ()){
                LogItem item = log . GetLogItem ();
```

```
                    for ( LogContent   content  :  item . GetLogCont
 ents ()){
                        System . out . print ( content . GetKey ()+":"+
 content . GetValue ());
                    }
                    System . out . println ();
                }
            }
        }
}
```

Precautions

1. To improve the I/O efficiency of your system, try not to directly use SDKs to write data to Log Service. For more information about the standard way to write data, see *Producer Library*.
2. To consume data in Log Service, try not to directly use SDKs to pull data interfaces. An advanced consumer library *Consumer group - Usage* is provided, which shields the implementation details of Log Service and provides the advanced functions such as load balancing and consumption in order.

3 .NET SDK

Download address

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

SDK GitHub address: *Click here to go to GitHub*

Procedure

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

Step 1 Create an Alibaba Cloud account

For more information, see *Sign up with Alibaba Cloud*.

Step 2 Obtain an Alibaba Cloud AccessKey

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

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

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

Step 3 Create a Log Service project and a Logstore

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

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

Note:

- Make sure that you use the same Alibaba Cloud account to obtain the Alibaba Cloud AccessKey and create the Log Service project and Logstore.

- For more information about the concepts of Log Service such as project and Logstore, see Core conceptBasic concepts.
- A project name must be globally unique in Log Service, and a Logstore name must be unique in the same project.
- After a project is created, you cannot modify the region or migrate the project across regions.

Step 4 Install the .NET development environment

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

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

Step 5 Download and install Log Service .NET SDK

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

1. Download
   - from GitHub: https://github.com/aliyun/aliyun-log-csharp-sdk
   - Historical version download: Clickhere to download the Log Service .NET SDK package
2. of the latest version. Log service. Net SDK is a software development kit that does not require additional installation. You can follow the steps below directly in your own visual studio projects.

Step 6 Start a new Log Service Net project

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

4 . Log Service SDK for .NET Core

Context

The Log Service SDK for .NET Core enables developers across different platforms to use Alibaba Cloud Log Service by means of the .NET Core framework.

Click here to download the Log Service SDK for .NET Core from GitHub.

Procedure

1. Create an Alibaba Cloud account

   For more information about how to create an Alibaba Cloud account, see *Sign up with Alibaba Cloud*.

2. Obtain an Alibaba Cloud AccessKey

   Before using the Log Service SDK for .NET Core, you must have obtained an *AccessKey*.

   Log on to your Alibaba Cloud account and go to the *Access Key Management page* . Then, select an AccessKey (consisting of an AccessKey ID and AccessKey Secret) for the SDK. If you do not have any AccessKeys, create one and make sure the AccessKey is enabled. For how to create an AccessKey, see *Preparation*.

   An enabled AccessKey is required for the following steps. Note that it must be kept confidential. For more information about how to use the AccessKey in the Log Service SDK for .NET Core, see *Configurations*.

3. Create a Log Service project and a Logstore

   Before using the Log Service SDK for .NET Core, you must have created a Log Service project and a Logstore in the console.

   For more information about how to create a project and a Logstore, see *Preparation*.

   > Note:
   >
   > - Ensure that you use the Alibaba Cloud account that obtained the AccessKey and created the project and the Logstore.
   > - For more information about concepts of Log Service (such as project and Logstore), see *Basic concepts*.

- A project name must be globally unique in the Log Service, and a Logstore name must be unique under the corresponding project.
- Once a project is created, you cannot modify the region or migrate the project across regions.

4. Install a .NET Core development environment

   Currently, the Log Service SDK for .NET Core supports the following versions:

   - . NET Core 2.0
   - . NET Framework (with . NET Core 1.x SDK) 4.6.2
   - . NET Framework (with . NET Core 2.0 SDK) 4.6.1

   For all supported versions, see *GitHub*.

5. Download and install the Log Store SDK for .NET Core.

   Build the .NET development environment and then download and install the Log Service SDK for .NET Core to the environment.

   a) Download .NET Core SDK.

      Click here to download the Log Service SDK for .NET Core from Github.

   b) Extract the installation package to the specified directory.

      The Log service SDK for .Net Core is a software development kit that does not require additional installation. To use the Log Service SDK for .NET Core in your Visual Studio project, follow these steps.

6. Create a new Log Service .Net Core project.

   Download the Log Service SDK for .NET Core and install it in the .NET development environment. Create a Log Service .NET Core project. For more information, see *Github Wiki*.

5 PHP SDK

Download address

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

Procedure

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

Step 1. Create an Alibaba Cloud account

For more information, see*Sign up with Alibaba Cloud* .

Step 2 Obtain an Alibaba Cloud AccessKey

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

Log on to the *Access Key Management page* . Select an AccessKey for for SDK. If you do not have any, create one and make sure the AccessKey is enabled. The AccessKey is used in the following steps and must be kept confidential. For more information about how to use the AccessKey in SDK, see*Preparation*SDK configuration.

The AccessKey is used in the following steps and must be kept confidential. See *Configurations* for more information about how to use the AccessKey in SDK.

Step 3. Create a Log Service project and a Logstore

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

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

Note:

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

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

Step 4 Install the PHP development environment

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

Step 5 Download and install PHP SDK

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

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

Step 6 Start a new PHP project

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

```
<? php
/* Use the autoloader class to automatica lly load all
   required PHP modules . Specify the proper path of
 the file containing the autoloader class .*/
 require_on ce realpath ( dirname ( __FILE__ ) . '/.. / Log_Autolo
 ad . php ');
$ endpoint = ' cn - hangzhou . sls . aliyuncs . com '; // Select
 the endpoint that matches the region of the project
 created above
$ accessKeyI d = ' your_acces s_key_id ';        // Use your
 Alibaba Cloud access key ID
$ accessKey = ' your_acces s_key ';             // Use your
 Alibaba Cloud access key secret
$ project = ' your_proje ct ';                  // The name of
   the project created in the above process
$ logstore = ' your_logst ore '; // The name of the
 Logstore created in the preceding step .
$ client = new Aliyun_Sls _Client ($ endpoint , $ accessId , $
 accessKey );
# List the names of all LogStores under the current
 project
$ request = new Aliyun_Sls _Models_Li stLogstore sRequest ($
 project );
$ response = $ client -> listLogsto res ($ request );
 var_dump ($ res1 );
# Create a LogStore
$ req2 = new Aliyun_Log _Models_Cr eateLogsto reRequest ($
 project ,$ logstore , 3 , 2 );
```

```
$ res2  = $ client  ->  createLogs  tore ($ req2 );
# Wait   until   the   LogStore   takes   effect
 sleep ( 60 );
# Write   logs
$ topic  = "";
$ source  = "";
$ logitems  =  array ();
 for  ($ i  =  0 ; $ i  <  5 ; $ i ++)
{
    $ contents  =  array (' index1 '=>  strval ($ i ));
    $ logItem  =  new   Aliyun_Log  _Models_Lo  gItem ();
    $ logItem -> setTime ( time ());
    $ logItem -> setContent  s ($ contents );
     Array_push  ($  logitems , $  logitem  );
}
$ req2  =  new   Aliyun_Log  _Models_Pu  tLogsReque  st ($ project , $
 logstore , $ topic , $ source , $ logitems );
$ res2  = $ client -> putLogs ($ req2 );
 var_dump ($ res4 );
#  Drag   data   immediatel  y
#  Traverse   shard   IDs
$ listShardR  equest  =  new   Aliyun_Log  _Models_Li  stShardsRe
 quest ($ project ,$ logstore );
$ listShardR  esponse  = $ client  ->  listShards ($ listShardR
 equest );
 foreach ($ listShardR  esponse ->  getShardId  s ()  as  $ shardId )
{
    # Obtain   the   cursor   correspond  ing   to   each   shard
 ID
    $ getCursorR  equest  =  new   Aliyun_Log  _Models_Ge  tCursorReq
 uest ($ project ,$ logstore ,$ shardId , null ,  time () -  60 );
    $ response  = $ client  ->  getCursor ($ getCursorR  equest );
    $ cursor  = $ response ->  getCursor ();
    $ count  =  100 ;
     while ( true )
    {
        # Read   data   starting   from   the   cursor .
        $ batchGetDa  taRequest  =  new   Aliyun_Log  _Models_Ba
 tchGetLogs  Request ($ project ,$ logstore ,$ shardId ,$ count ,$
 cursor );
         var_dump ($ batchGetDa  taRequest );
        $ response  = $ client  ->  batchGetLo  gs ($ batchGetDa
 taRequest );
         if ($ cursor  == $ response  ->  getNextCur  sor ())
        {
             break ;
        }
        $ logGroupLi  st  = $ response  ->  getLogGrou  pList ();
         foreach ($ logGroupLi  st   as  $ logGroup )
        {
             print  ($ logGroup -> getCategor  y ());
             foreach ($ logGroup  ->  getLogsArr  ay ()  as  $ log )
            {
                 foreach ($ log  ->  getContent  sArray ()  as  $
 content )
                {
                     print ($ content ->  getKey ().":".$ content ->
 getValue ()."\ t ");
                }
                 print ("\ n ");
            }
        }
        $ cursor  = $ response  ->  getNextCur  sor ();
    }
```

```
}
# Wait   one   minute   until   logs   can   be   queried .
 sleep ( 60 );
# Query   log   distributi  on  ( NOTE :  Ensure   that   indexes
 are   created   before   you   query   logs .  The   PHP   SDK   does
   not   provide   this   interface ,  so   you   need   to   create
 it   on   the   console .)
$ topic  = "";
$ query ='';
$ from  =  time ()- 3600 ;
$ to  =  time ();
$ res3  =  NULL ;
 while  ( is_null ($ res3 ) || (! $ res3 -> isComplete  d ()))
{
    $ req3  =  new   Aliyun_Log  _Models_Ge  tHistogram  sRequest ($
 project , $ logstore , $ from , $ to , $ topic , $ query );
    $ res3  = $ client -> getHistogr  ams ($ req3 );
}
 var_dump ($ res3 );
# Query   log   data .
$ res4  =  NULL ;
 while  ( is_null ($ res4 ) || (! $ res4 -> isComplete  d ()))
{
    $ req4  =  new   Aliyun_Log  _Models_Ge  tLogsReque  st ($ project
 , $ logstore , $ from , $ to , $ topic , $ query ,  5 ,  0 ,  False
 );
     res4  =  client . GetLogs ( req4 );
}
 var_dump ($ res4 );
```

6 Python SDK

Download address

SDK GitHub:

https://github.com/aliyun/aliyun-log-python-sdk

Procedure

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

Step 1. Create an Alibaba Cloud account

For more information, see *Sign up with Alibaba Cloud*.

Step 2. Obtain an Alibaba Cloud AccessKey

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

Log on to the *Access Key Management page* . Select an AccessKey for SDK. If you do not have any, create one and make sure the AccessKey is enabled. For more information about how to create an access key, see *Preparation*.

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

Step 3 Create a Log Service project and a Logstore

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

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

1. Log on to the Log Service console.
2. Click Create Project in the upper-right corner.Click Create Project in the upper right corner.
3. Enter the Project Name and select the Region. Click Confirm.
4. On the Project List page, click the name of the project, and then click Create. *Create a Logstore*.

   After you create a project, you can also click Create to create a Logstore based on the system prompt.

5. Complete the configurations, and click Confirm.

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

   

   Note:

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

Step 4. Install a Python environment

The Python SDK is a pure Python library and supports all Python operating systems, including Linux, Mac OS X and Windows. Please install Python as follows:

1. Download and install the latest Python *installation package*.

   

   Note:

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

2. Download and install the Python package management tool *pip*.

   After pip is installed, run `pip -V` to check whether the installation is successful and query the current pip version.

Step 5. Install a Python SDK

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

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

Step 6. Start a Python program

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

For more information, see *Github/readthedocs*.

```
# encoding: utf-8
import time
from aliyun.log.logitem import LogItem
from aliyun.log.logclient import LogClient
from aliyun.log.getlogsrequest import GetLogsRequest
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
from aliyun.log.gethistogramsrequest import GetHistogramsRequest
def main():
    endpoint = '' # Select the endpoint that matches the region of the project created in the preceding step.
    accessKeyId = '' # Use your Alibaba Cloud AccessKey ID.
    accessKey = '' # Use your Alibaba Cloud AccessKey Secret.
    project = '' # The name of the project created in the preceding step.
    logstore = '' # The name of the Logstore created in the preceding step.
    # Note: Configure four shards for the created Logstore for later testing.
    # Construct a client
    client = LogClient(endpoint, accessKeyId, accessKey)
    # List all LogStores
    req1 = ListLogstoresRequest(project)
    res1 = client.list_logstores(req1)
    res1.log_print()
    topic = ""
    source = ""
    # Send 10 data packets, each of which contains 10 logs.
    for i in range(10):
        logitemList = [] # LogItem list
        for j in range(10):
            contents = [('index', str(i*10+j))]
            logItem = LogItem()
            logItem.set_time(int(time.time()))
            logItem.set_contents(contents)
            logitemList.append(logItem)
        req2 = PutLogsRequest(project, logstore, topic, source, logitemList)
        res2 = client.put_logs(req2)
        res2.log_print()
```

```
    # List all shards and read the data written in
the last minute .
    listShardR es = client . list_shard s ( project , logstore )
    for shard in listShardR es . get_shards _info ():
        shard_id = shard [" shardID "]
        start_time = int ( time . time () - 60 )
        end_time = start_time + 60
        res = client . get_cursor ( project , logstore , shard_id
, start_time )
        res . log_print ()
        start_curs or = res . get_cursor ()
        res = client . get_cursor ( project , logstore , shard_id
, end_time )
        end_cursor = res . get_cursor ()
        while True :
            loggroup_c ount = 100 # Read 100 packets each
 time .
            res = client . pull_logs ( project , logstore ,
shard_id , start_curs or , loggroup_c ount , end_cursor )
            res . log_print ()
            next_curso r = res . get_next_c ursor ()
            if next_curso r == start_curs or :
                break
            start_curs or = next_curso r
    # Note : You can use the following interfaces to
query data only when the index function is enabled .
    time . sleep ( 60 )
    topic = ""
    query = " index "
    From = int ( time . time ()) - 600
    To = int ( time . time ())
    res3 = None
    # Query the number of logs that match the query
 criteria during the past 10 minutes . Retry if not
all execution results are correct .
    while ( res3 is None ) or ( not res3 . is_complet ed
()):
        req3 = GetHistogr amsRequest ( project , logstore , From
, To , topic , query )
        res3 = client . get_histog rams ( req3 )
    res3 . log_print ()
    # Obtain the number of logs that match the query
 conditions .
    total_log_ count = res3 . get_total_ count ()
    log_line = 10
    # Read 10 logs each time until all log data is
 queried . Retry three times if not all query results
 are correct during each query .
    for offset in range ( 0 , total_log_ count , log_line ):
        res4 = None
        for retry_time in range ( 0 , 3 ):
            req4 = GetLogsReq uest ( project , logstore , From
, To , topic , query , log_line , offset , False )
            res4 = client . get_logs ( req4 )
            if res4 is not None and res4 . is_complet ed
():
                break
            time . sleep ( 1 )
        if res4 is not None :
            res4 . log_print ()
    listShardR es = client . list_shard s ( project , logstore )
    shard = listShardR es . get_shards _info ()[ 0 ]
    # Split a shard
    if shard [" status "] == " readwrite ":
```

```
        shard_id  =  shard [" shardID "]
        inclusiveB  eginKey  =  shard [" inclusiveB  eginKey "]
        midKey  =  inclusiveB  eginKey [:- 1 ] +  str (( int (
inclusiveB  eginKey [- 1 :])) +  1 )
        client . split_shar  d ( project ,  logstore ,  shard_id ,
midKey )
   #  Merge   shards .
    shard  =  listShardR  es . get_shards  _info ()[ 1 ]
    if   shard [" status "] == " readwrite ":
        shard_id  =  shard [" shardID "]
        client . merge_shar  d ( project ,  logstore ,  shard_id )
   #  Delete   shard
    shard  =  listShardR  es . get_shards  _info ()[- 1 ]
    if   shard [" status "] == " readonly ":
        shard_id  =  shard [" shardID "]
        client . delete_sha  rd ( project ,  logstore ,  shard_id )
  #  Create   an   external   store .
    res  =  client . create_ext  ernal_stor  e ( project ,
ExternalSt  oreConfig (" rds_store "," cn - qingdao "," rds - vpc ","
vpc -************","  i ***********","*. *. *.*"," 3306 "," root ","
sfdsfldsfk  sflsdfs "," meta "," join_meta "));
    res . log_print ()
    res  =  client . update_ext  ernal_stor  e ( project ,
ExternalSt  oreConfig (" rds_store "," cn - qingdao "," rds - vp ","
rds - vpc "," vpc -************","  i ***********","*. *. *.*"," 3306
"," root "," sfdsfldsfk  sflsdfs "," meta "," join_meta "));
    res . log_print ()
    res  =  client . get_extern  al_store ( project ," rds_store ");
    res . log_print ()
    res  =  client . list_exter  nal_store ( project ,"");
    res . log_print ();
    res  =  client . delete_ext  ernal_stor  e ( project ," rds_store
")
    res . log_print ();
   #  Use   python   sdk   for   query   analysis .
    req4  =  GetLogsReq  uest ( project ,  logstore ,  From ,  To ,
topic , "* |  select   count ( 1 )",  10 , 0 ,  False )
    res4  =  client . get_logs ( req4 )
   #  Use   python   sdk   for   join   rds   query .
    req4  =  GetLogsReq  uest ( project ,  logstore ,  From ,  To ,
topic , "* |  select   count ( 1 )  from  "+ logstore  +"  l   join
rds_store   r   on   l . ikey  = r . ekey ",  10 , 0 ,  False )
    res4  =  client . get_logs ( req4 )
   #  Use   python   sdk   to   insert   query   results   into   rds
.
    req4  =  GetLogsReq  uest ( project ,  logstore ,  From ,  To ,
topic , "* |  insert   into   rds_store   select   count ( 1 ) ",  10
, 0 ,  False )
    res4  =  client . get_logs ( req4 )
if   __name__  == ' __main__ ':
    main ()
```

7 Android SDK

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

GitHub address:

Https://github.com/aliyun/aliyun-log-android-sdk

8 C SDK

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

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

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

For the GitHub project addresses, see:

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

9 Go

Alibaba Cloud Log Service Go SDK supports:

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

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

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

10 iOS SDK

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

GitHub: *https://github.com/aliyun/aliyun-log-ios-sdk*

Swift

```
 /*
     Use   the   endpoint ,  AccessKey   ID ,  and   AccessKey
 Secret   to   build   the   Log   Service   client .
    @ endPoint :  see   https :// www . alibabaclo  ud . com / help /
 doc - detail / 29008 . htm
 */
 let   myClient  =  try !  LOGClient ( endPoint : "",
                              accessKeyI  D : "",
                              accessKeyS  ecret : "",
                              projectNam  e :"")
/*  Create   a   log   group . */
 Let   loggroup  =  try !  Loggroup  ( topic : " mtopic ",  source : "
 msource  ")
      /*  Store   a   log . */
     let   log1  =  Log ()
           try !  log1 .  PutContent (" K11 ",  value : " V11 ")
          try !  log1 .  PutContent (" K12 ",  value : " V12 ")
          try !  log1 .  PutContent (" K13 ",  value : " V13 ")
     logGroup . PutLog ( log1 )
   /*  Store   a   log . */
     Let   log2  =  Log  ()
           try !  log2 .  Putcontent  (" k21 ",  value : " V21  ")
          try !  log2 .  PutContent (" K22 ",  value : " V22 ")
          try !  log2 .  PutContent (" K22 ",  value : " V22 ")
     logGroup . PutLog ( log2 )
  /*  Send   the   log . */
  myClient . PostLog ( logGroup , logStoreNa  me : ""){  response ,
 error   in
        //  handle   response   however   you   want
         if   error ?. domain  ==  NSURLError  Domain  &&  error ?.
 code  ==  NSURLError  TimedOut  {
             print (" timed   out ") //  note , ` response `  is
 likely  ` nil `  if   it   timed   out
        }
    }
```

Objective-C

See GitHub: *https://github.com/lujiajing1126/AliyunLogObjc*

11 C++ SDK

You can use the Alibaba Cloud Log Service C++ SDK to call Log Service APIs for servers running Linux.

The Alibaba Cloud Log Service C++ SDK supports all Log Service APIs and provides various functions, such as resource creation and data read/write.

You can obtain the C++ SDK from GitHub at *https://github.com/aliyun/aliyun-log-cpp-sdk*.