Alibaba Cloud Application Configuration Management

API Reference

Issue: 20190118

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 intellectu

al 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 instructio ns, 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]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 API overview

This section describes the APIs that you can use in application configuration service.

Currently, configuration query API and listening-related APIs are available. Before using these APIs, make sure you have carefully read the ACM product documentation and accepted the user agreement.

2 How to request ACM services

This topic explains how to request ACM services.

Domain name of the address servers

Region	Domain name of the address server
Internet	acm.aliyun.com
China (Hangzhou)	addr-hz-internal.edas.aliyun.com
China (Qingdao)	addr-qd-internal.edas.aliyun.com
China (Shanghai)	addr-sh-internal.edas.aliyun.com
China (Beijing)	addr-bj-internal.edas.aliyun.com
China (Shenzhen)	addr-sz-internal.edas.aliyun.com
China (Hong Kong)	addr-hk-internal.edas.aliyuncs.com
Singapore	addr-singapore-internal.edas.aliyun.com
Australia (Sydney)	addr-ap-southeast-2-internal.edas.aliyun.com
US (Silicon Valley)	addr-us-west-1-internal.acm.aliyun.com
US (Virginia)	addr-us-east-1-internal.acm.aliyun.com
China (Shanghai) Financial Cloud	addr-cn-shanghai-finance-1-internal.edas.aliyun.com

Get ACM server list

Retrieve the IP addresses of ACM server through Address Server, so that you can get configurations by sending requests to the server IP.

```
http://${Address_Server_Domain}:8080/diamond-server/diamond
```

For example:

```
curl http://acm.aliyun.com:8080/diamond-server/diamond
139.196.135.144
```

Communication protocols

Supports request communication using HTTP.

Request method

Allows sending HTTP GET or POST requests. In the HTTP GET request, the parameters must be included in the request URL.

Request parameters

Each request must contain the request parameters related to public authentication and signatures, as well as the specific operation-related parameters.

Character encoding

Both requests and returned results are encoded using the GBK character set.

Signature mechanism

The ACM service performs authentication on each access request. Therefore, each request being sent over HTTP protocol must contain signature information. By using the AccessKey and SecretKey, the ACM performs symmetric encryption to authenticate the request sender.

The accessKey and the secretKey are issued to the visitor by ACM. The accessKey is used for authenticating the visitor, and the secretKey is the key that encrypts the signature string and then validates it on the server. Only you and ACM know them, and they must remain strictly confidential.

Signature algorithm

The HMACSHA1 algorithm is used for signing. The following describes the examples of a Java and Shell signature algorithms.

- Example of a Java signature algorithm

```
public static void main(String[] args) throws Exception {
    String tenant= "tenant";
    String group = "group";
    String timeStamp = String.valueOf(System.currentTimeMillis());
    String abc = HmacSHA1Encrypt(tenant+ "+" + group + "+" +
timeStamp , "1234");
    System.out.println(abc);
}
public static String HmacSHA1Encrypt(String encryptText, String
encryptKey) throws Exception {
    byte[] data = encryptKey.getBytes("UTF-8");
    // Construct a key based on the given byte array and specify the
 name of a key algorithm in the second parameter.
    SecretKey secretKey = new SecretKeySpec(data, "HmacSHA1");
    // Generate a Mac object for the specified Mac algorithm
    Mac mac = Mac.getInstance("HmacSHA1");
    // Initialize the Mac object with the given key
    mac.init(secretKey);
    byte[] text = encryptText.getBytes("UTF-8");
    byte[] textFinal = mac.doFinal(text);
```

```
    // Complete the Mac operation and Base64 encoding. Convert the
byte array to a string.
    return new String(Base64.encodeBase64(textFinal));
}
```

- Shell signature algorithm

```
## config sign
timestamp=`echo $[$(date +%s%N)/1000000]`
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $sk -sha1 -binary
 | base64`
echo $signContent
```

Signature procedure

1. Use request parameters to construct a canonicalized query string (QueryParam).
2. Follow the subsequent rules to construct the string for signature calculation using the canonicalized query string constructed in the previous step.

```
Signature=
 HMAC-SHA1(QueryParam)
```

Note:

The QueryParam varies with requests.

3. Use the preceding signature sting to calculate the signature's HMAC value based on RFC2104 definitions. Note: The key used for signature calculation is the Access Key Secret held by the user (ASCII:38), and the hash algorithm used is SHA1.
4. According to Base64 encoding rules, encode the preceding HMAC value, which gives you the signature value.
5. Add the obtained signature value to the request parameters as the "Signature" parameter, which completes the request signing process.

Code examples

This is an example of constructing an ACM request with Shell.

```
#! /bin/bash
## config param
dataId="xxx"
group="xxx"
namespace="xxx"
accessKey="xxx"
secretKey="xxx"
endpoint="xxx"
## config param end
## get serverIp from address server
serverIp=`curl $endpoint:8080/diamond-server/diamond -s | awk '{a[NR]=
$0}END{srand();i=int(rand()*NR+1);print a[i]}'`
## config sign
timestamp=`echo $[$(date +%s%N)/1000000]`
```

```
signStr=$namespace+$group+$timestamp
signContent=`echo -n $signStr | openssl dgst -hmac $secretKey -sha1 -
binary | base64`
## request to get a config
curl -H "Spas-AccessKey:"$accessKey -H "timeStamp:"$timestamp -H "
Spas-Signature:"$signContent "http://"$serverIp":8080/diamond-server/
config.co?dataId="$dataId"&group="$group"&tenant="$namespace -v
```

3 Get configurations

This topic explains how to use API to get configurations from ACM.

Description

It gets configurations from ACM.

Request Type

GET

Request URL

`/diamond-server/config.co`

Request parameters

Parameter	Type	Required	Description
tenant	String	Yes	The tenant, corresponding to the namespace field of ACM
dataId	String	Yes	Configuration ID
group	String	Yes	Configuration group

Header parameters

Name	Type	Required	Description
Spas-AccessKey	String	Yes	The accessKey can be found in the ACM console.
timeStamp	String	Yes	The request time in milliseconds
Spas-Signature	String	Yes	`SpasSigner.sign(Tenant+ group+ timeStamp , secretKey)`. Sign "tenant + group + timestamp" with secret key. The signature algorithm is HmacSHA1. The timestamp signature prevents replay attacks. The signature is valid for 60 seconds.
Spas-SecurityToken	String	No	SecurityToken is obtained from STS temporary credential. STS temporary credential is obtained from instance metadata URL. For more information, see: • *Access other Cloud Product APIs by the Instance RAM Role* • *Access ACM with instance RAM role*

Response parameters

Parameter Type	Description
String	Configuration value

Error code

Error code	Error message	Explanation
400	Bad Request	Syntax error in client request
403	Forbidden	No permission
404	Not Found	Client error, not found
500	Internal Server Error	Internal errors of the server
200	OK	Normal

Examples

- Request example

```
http:serverIp:8080/diamond-server/config.co? dataId=dataIdparam&
group=groupParam&tenant=tenantParam
```

- Response example

```
contentTest
```

4 Get the configurations within a namespace

This topic explains how to use API to get configurations from ACM.

Description

It gets the configurations within a namespace from ACM.

Request type

GET

Request URL

`/diamond-server/basestone.do? method=getAllConfigByTenant`

Request parameters

Parameter	Type	Required	Description
tenant	String	Yes	The tenant, corresponding to the namespace field of ACM
pageNo	int	Yes	Page number
pageSize	int	Yes	Page size

Header parameters

Name	Type	Required	Description
Spas-AccessKey	String	Yes	The accessKey can be found in the ACM console.
timeStamp	String	Yes	The request time in milliseconds
Spas-Signature	String	Yes	`SpasSigner.sign(Tenant+ group+ timeStamp , secretKey)`. Sign "tenant + group + timestamp" with secret key. The signature algorithm is HmacSHA1. The timestamp signature prevents replay attacks. The signature is valid for 60 seconds.
Spas-SecurityToken	String	No	SecurityToken is obtained from STS temporary credential. STS temporary credential is obtained from instance metadata URL. For more information, see: • *Access other Cloud Product APIs by the Instance RAM Role* • *Access ACM with instance RAM role*

Response parameters

Parameter type	Description
totalCount	The total number of configurations
pageNumber	Page number
pagesAvailable	The number of available pages
pageItems	Configuration items

Error code

Error code	Error message	Explanation
400	Bad Request	Syntax error in client request
403	Forbidden	No permission
404	Not Found	Client error, not found
500	Internal Server Error	Internal errors of the server
200	OK	Normal

Examples

- Request example

```
http:serverIp:8080/diamond-server/basestone.do? method=getAllConf
igByTenant&amp;tenant=tenantParam&amp;pageNumber=pageNumberParam&amp
;pageSize=pageSizeParam
```

- Response example

```
contentTest
```

5 Listen to configurations

This topic explains how to use API to listen to configurations in ACM.

Description

Capture configuration changes in real time by listening to configurations on ACM. In case of any configuration changes, you can use the *Get configurations* to obtain the latest value of the configuration and dynamically refresh the local cache.

Registered listening adopts asynchronous Servlet technique. Registered listening is essentially to compare the MD5 with configurations and configuration values to that in the backend. If the MD5 values don't match, then it immediately returns the different configuration. If they match, it holds for 30 seconds. And it returns an empty string.

Request type

POST

Request URL

/diamond-server/config.co

Request parameters

Parameter	Type	Required	Description
Probe-Modify-Request	String	Yes	Listen to data messages. The format is `dataId^2group^2contentMD5^2tenant^1`. • dataId: Configuration ID • group: Configuration group • contentMD5: MD5 of the configuration content • tenant: The tenant, corresponding to the namespace field of ACM

Header parameters

Parameter	Type	Required	Description
longPullingTimeout	String	Yes	If the long pulling timeout is 30 seconds, then enter 30000 here.
Spas-AccessKey	String	Yes	The accessKey can be found in the ACM console.
timeStamp	String	Yes	The request time in milliseconds

Parameter	Type	Required	Description
Spas-Signature	String	Yes	`SpasSigner.sign(Tenant+ group+ timeStamp , secretKey)`. Sign "tenant + group + timestamp" with secret key. The signature algorithm is HmacSHA1. The timestamp signature prevents replay attacks. The signature is valid for 60 seconds.

Parameter description

- A delimiter to separate fields within a configuration: `^2 = Character.toString((char) 2`
- A delimiter to separate configurations: `^1 = Character.toString((char) 1)`
- contentMD5: MD5(content). This is an empty string because the first local cache is empty.

Response parameters

Parameter type	Description
String	Configuration value

Error code

Error code	Error message	Explanation
400	Bad Request	Syntax error in client request
403	Forbidden	No permission
404	Not Found	Client error, not found
500	Internal Server Error	Internal errors of the server
200	OK	Normal

Examples

- Request example

```
http://serverIp:8080/diamond-server/config.co
POST request body data:
Probe-Modify-Request=dataId^2group^2contentMD5^2tenant^1
```

- Response example

```
In case of any configuration changes
dataId^2group^2tenant^1
If no configuration changes: an empty string is returned
```

6 Publish configurations

This topic explains how to use API to publish configurations to ACM.

Description

It publishes configurations to ACM.

Request type

POST

Request URL

/diamond-server/basestone.do

Request parameters

Name	Type	Required	Description
tenant	String	Yes	The tenant, corresponding to the namespace field of ACM
dataId	String	Yes	Configuration ID
group	String	Yes	Configuration group
content	String	Yes	The content of the configuration

Header parameters

Name	Type	Required	Description
Spas-AccessKey	String	Yes	The accessKey can be found in the ACM console.
timeStamp	String	Yes	The request time in milliseconds
Spas-Signature	String	Yes	`SpasSigner.sign(Tenant+ group+ timeStamp , secretKey)`. Sign "tenant + group + timestamp" with secret key. The signature algorithm is HmacSHA1. The timestamp signature prevents replay attacks. The signature is valid for 60 seconds.
Spas-SecurityToken	String	No	SecurityToken is obtained from STS temporary credential. STS temporary credential is obtained from instance metadata URL. For more information, see: • *Access other Cloud Product APIs by the Instance RAM Role* • *Access ACM with instance RAM role*

Response parameters

Parameter Type	Description
boolean	If the publishing is successful

Error code

Error code	Error message	Explanation
400	Bad Request	Syntax error in client request
403	Forbidden	No permission
404	Not Found	Client error, not found
500	Internal Server Error	Internal errors of the server
200	OK	Normal

Examples

- Request example

```
http:serverIp:8080/diamond-server/basestone.do? method=syncUpdateAll
http body:
dataId=dataIdparam&group=groupParam&tenant=tenantParam&content=
contentParam
```

- Response example

```
true
```

7 Delete configurations

This topic explains how to use API to delete configurations from ACM.

Description

It deletes configurations from ACM.

Request type

POST

Request URL

`/diamond-server/datum.do`

Request parameters

Parameter	Type	Required	Description
tenant	String	Yes	The tenant, corresponding to the namespace field of ACM
dataId	String	Yes	Configuration ID
group	String	Yes	Configuration group

Header parameters

Name	Type	Required	Description
Spas-AccessKey	String	Yes	The accessKey can be found in the ACM console.
timeStamp	String	Yes	The request time in milliseconds
Spas-Signature	String	Yes	`SpasSigner.sign(Tenant+ group+ timeStamp , secretKey)`. Sign "tenant + group + timestamp" with secret key. The signature algorithm is HmacSHA1. The timestamp signature prevents replay attacks. The signature is valid for 60 seconds.
Spas-SecurityToken	String	No	SecurityToken is obtained from STS temporary credential. STS temporary credential is obtained from instance metadata URL. For more information, see: • *Access other Cloud Product APIs by the Instance RAM Role* • *Access ACM with instance RAM role*

Response parameters

Parameter type	Description
boolean	If the deletion is successful

Error code

Error code	Error message	Explanation
400	Bad Request	Syntax error in client request
403	Forbidden	No permission
404	Not Found	Client error, not found
500	Internal Server Error	Internal errors of the server
200	OK	Normal

Examples

- Request example

```
http:serverIp:8080/diamond-server/datum.do? method=deleteAllDatums
http body:
dataId=dataIdparam&group=groupParam
```

- Response example

```
true
```