ALIBABA CLOUD

阿里云

云服务总线 CSB
开发指南

文档版本：20210118

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.SDK参考

CSB提供Java版本的HTTP Client SDK和WebService Client SDK。

获取SDK

目前CSB SDK已经开源，发布地址为*https://github.com/aliyun/csb-sdk*，您可以获取相关的代码并了解SDK的详细信息。

您还可以查看CSB SDK的版本信息。

HTTP SDK

HTTP SDK用来调用由CSB发布的HTTP服务，它主要用来向服务端发送HTTP请求，请求调用支持POST和GET方式。如果提供了AccessKey ID和AccessKey Secret参数信息，它能够在内部将请求消息进行签名处理，然后向CSB服务端发送请求信息进行验证和调用。

> 注意
>
> - 该版本的SDK要求的运行环境为JDK 1.7或8以上版本。
> - 该版本的SDK支持的CSB版本为1.1.5.x。
> - 不再建议使用1.1.5.7之前的版本。如果您仍在使用，建议升级到最新版本的 SDK。

HTTP SDK的下载地址如下：

- http-client-1.1.5.8（最新）
- http-client-1.1.5.7

文档地址（不定期更新）：HTTP SDK 使用说明

- 介绍了HTTP SDK的使用方式，帮助CSB OpenAPI的使用者了解如何使用SDK进行命令行或者编程方式调用该API。
- 介绍了HTTP SDK的签名机制，帮助非Java语言的开发者了解CSB-HTTP请求的签名原理以便编写其它语言的SDK实现。
- 提供了除Java版本的SDK之外，其它几种开发语言的SDK参考实现。

WebService SDK

WebService SDK用来调用由CSB发布的WebService服务，它主要用来在每次调用时做方法拦截把安全需要的KV信息添加到HTTP请求头部分。

> 说明
>
> - 该版本的SDK要求的运行环境为JDK 1.7或8以上版本。
> - 该版本的SDK支持的CSB版本为1.1.5.x。
> - 不再建议使用1.1.5.7之前的版本。如果您仍在使用，建议升级到最新版本的 SDK。

WebService SDK的下载地址如下：

- ws-client-1.1.5.8（最新）
- ws-client-1.1.5.7

文档的地址（不定期更新）：WebService SDK 使用说明

- 介绍了WebService SDK的使用方式，帮助CSB OpenAPI的使用者了解如何使用SDK进行命令行或者编程方式调用该API。
- 介绍了WebService SDK的签名机制，帮助非Java语言的开发者了解CSB-WebService请求的签名原理以便编写其它语言的SDK实现。

使用SDK示例

示例下载地址：sdk-sample.zip

- 通过编程方式使用HTTP SDK
- 通过编程方式使用WebService SDK

SDK的使用示例，请参见SDK 使用示例的说明。

定制扩展能力

CSB SDK支持以下定制扩展能力：

- 自定义定制扩展能力。
- 支持插件式定制实现自定义的验签逻辑、流量控制、预请求处理和响应处理。

更多信息，请参见CSB 用户自定义扩展说明。

历史版本支持

> **注意** 不再建议使用1.1.5.7之前的版本。如果您仍在使用，建议升级到最新版本的SDK。

- 1.0.4.1
  - Java 1.7
  - Java 1.6
  - 示例
- 1.0.4.4

  使用工具完成服务的导入导出，请参见服务批量导入导出工具说明。

  > **说明**
  > - 此功能可以直接使用最新版的HTTP-SDK完成。
  > - 此批量导入和导出功能已经可以在控制台界面中操作。

2.使用HTTP-SDK调用Console API

使用HTTP-Client SDK可以对CSB控制台的OpenAPI进行调用，帮助您完成开发和运维操作。

前提条件

- 下载HTTP-SDK。
  - http-client-1.1.5.8（最新）
  - http-client-1.1.5.7
- 按照OpenAPI的文档了解该API的调用地址、请求参数、请求方式（POST和GET）和返回值的结构。
- 确定要调用API的安全Key信息，即管理员用户创建凭证的一对有效的AccessKey ID和AccessKey Secret。

背景信息

HTTP-SDK介绍和使用方式，请参见SDK参考。使用HTTP-SDK调用Console API有命令行和代码两种方式。

使用命令行调用

命令行方式通常用来测试或临时调用OpenAPI。

命令行参数说明：

```
#java -jar http-client-1.1.5.8.jar
usage: java -jar http-client.jar [options...]
-method <arg>              请求类型, 默认get, 可选的值为: get, post, cget和cpost。
-url <arg>                 测试:请求地址，例如http://CSB服务地址:8086/CSB?p1=v1。CSB服务地址即创建该实例时绑定的SLB的地址。
-api <arg>                 服务名。
-version <arg>             服务版本。
-ak <arg>                  AccessKey ID, 可选。
-sk <arg>                  AccessKey Secret, 可选。
-D <arg>                   请求参数, 格式: -D "key=value" 可以定义多个-D参数。
-H <arg>                   HTTP Header, 格式: -H "key:value" 可以定义多个-H参数。
-cbJSON <arg>              以JSON串方式POST发送的请求body, 例如: -cbJSON "{'name':'wiseking'}"。
-cc,--changeCharset        返回值是否需要转换charset。
-h,--help                  打印帮助信息。
-d,--debug                 打印调试信息。
-nonce                     是否做nonce防重放处理，不定义为不做nonce重放处理。
-proxy <arg>               设置代理地址, 格式: proxy_hostname:proxy_port。
-sdkv,--sdk-version        SDK版本信息。
-sign,--signImpl <arg>     客户端签名类, 可选。
-verify,--verifySignImpl <arg>  CSB服务端验签类, 可选。
```

下面介绍使用SDK命令行调用API的示例。

- 获取实例列表

```
java -jar http-sdk.jar \
-method get \
-api /api/csbinstance/listCsbs \
-version 1.1.0.0 \
-ak replace-ak \
-sk replace-sk \
-url https://csb.console.aliyun.com/api/csbinstance/listCsbs
```

- 在某个实例中创建一个新的服务组

```
java -jar http-sdk.jar \
-method get \
-api /api/project/createorupdate \
-version 1.1.0.0 \
-ak replace-ak \
-sk replace-sk \
-Ddfile=dfile.prop \
-url https://csb.console.aliyun.com/api/project/createorupdate?csbId=175
```

其中dfile.prop的内容如下：

```
data={"description":"openapi test2","projectName":"lt-测试服务组"}
```

> **注意**
> - API定义的是接口全路径，必须以“/”开头。
> - 必须根据API的要求使用正确的方法（Method）。

使用代码调用

> **说明** 需要在应用的*pom.xml*中添加http-client-sdk的依赖。

下面介绍使用代码调用API的示例。

- 获取实例列表

```
import com.alibaba.csb.sdk.HttpCaller;
import com.alibaba.csb.sdk.HttpCallerException;
import com.alibaba.csb.sdk.HttpParameters;
import com.alibaba.fastjson.JSON;
public class ConsoleAPI {
  public static void main(String[] args) throws HttpCallerException {
    listCsbInstance();
  }
  public static void listCsbInstance() throws HttpCallerException {
    String requestURL = "http://csb.console.1120.daily.server"; //console域名。
    String apiName = "/api/csbinstance/listCsbs";
    String queryParams = "";
    String version = "1.1.2.0";
    String method = "get";
    String ak = "ac46xxxx82c64xxxx2c0axxxx4b82xxx";
    String sk = "jonvxxxxGeKVxxxxKpxNxxxx6uxx";
    HttpParameters.Builder hp = HttpParameters.newBuilder()
        .requestURL(requestURL + apiName + queryParams)
        .api(apiName)
        .version(version)
        .accessKey(ak)
        .secretKey(sk)
        .method(method);
    hp.putParamsMap("pageNum", "2");
    String ret = HttpCaller.invoke(hp.build());
    System.out.println("ret=" + JSON.toJSONString(JSON.parse(ret), true));
  }
}
```

- 在某个实例中创建一个新的服务组

```
import com.alibaba.csb.sdk.HttpCaller;
import com.alibaba.csb.sdk.HttpCallerException;
import com.alibaba.csb.sdk.HttpParameters;
import com.alibaba.fastjson.JSON;
public class ConsoleAPI {
  public static void main(String[] args) throws HttpCallerException {
    createProject();
  }
  public static void createProject() throws HttpCallerException {
    String apiName = "/api/project/createorupdate"; // 请求接口。
    String version = "1.1.2.0";
    String method = "post";
    String ak = "ac46xxxx82c64xxxx2c0axxxx4b82xxx";
    String sk = "jonvxxxxGeKVxxxxKpxNxxxx6uxx";
    String data = "{\"projectName\":\"CSB服务组测试\"" +
        ",\"projectOwnerName\":\"CSBTest\"" +
        ",\"projectOwnerEmail\":\"\"" +
        ",\"projectOwnerPhoneNum\":\"\"" +
        ",\"description\":\"\"}";
    HttpParameters.Builder hp = HttpParameters.newBuilder()
        .requestURL(CSB_CONSOLE_DOMAIN + apiName + "?csbId=176") // 请求地址可以附带请求参数。
        .api(apiName)
        .version(version)                  // 设置调用的API和版本。
        .accessKey(ak).secretKey(sk)       // 设置使用当前实例账号的AccessKey ID和AccessKey Secret。
        .method(method)                    // 设置调用方式。
        .putParamsMap("data", data);       // 设置请求的form body参数。
    String ret = HttpCaller.invoke(hp.build());     // 进行调用。
    System.out.println("retStr =" + ret);
  }
}
```

3.管理API

管理API是CSB提供给用户和第三方开发者使用的CSB管理接口。

API访问

- 访问地址

  目前已部署5个地域，后续会根据需要开通其它地域的管理API。各个区域的访问地址如下表：

地域	访问地址
华东1（杭州）	csb.cn-hangzhou.aliyuncs.com
华东2（上海）	csb.cn-shanghai.aliyuncs.com
华北2（北京）	csb.cn-beijing.aliyuncs.com
华南1（深圳）	csb.cn-shenzhen.aliyuncs.com
中国香港	csb.cn-hongkong.aliyuncs.com

- 访问权限

  目前只开通白名单用户有权访问管理API，需要使用此管理API的用户，请您联系CSB接口人。

- 流量控制

  目前配置和管理API的流量控制：每API 50 tpm、每用户每API 5 tpm。

  > **说明** 如果此流控阈值不满足您的要求，请您联系CSB接口人。

- API访问方法

  请使用最新的1.2.8版本的管理API SDK。

  此SDK是访问CSB管理API的SDK，不能用于访问CSB上发布的业务服务。CSB上的业务服务访问，请参见SDK参考。

Java SDK

使用方法，请参见阿里云 SDK 说明 。

CSB管理API SDk的Maven依赖（详细信息可查看Maven 仓库）：

```
<dependency>
 <groupId>com.aliyun</groupId>
 <artifactId>aliyun-java-sdk-csb</artifactId>
 <version>1.2.8</version>
</dependency>
```

示例代码：

```
public static void main(String[] args) {
    try {
      // 创建DefaultAcsClient实例并初始化，设置对应regoin的endPoint。
      DefaultProfile.addEndpoint("CSB", "cn-hangzhou", "CSB", "csb.cn-hangzhou.aliyuncs.com");
      DefaultProfile profile = DefaultProfile.getProfile(
          "cn-hangzhou",       // 地域。
          "****",    // RAM用户的AccessKey ID。
          "****"); // RAM用户的AccessKey Secret。
      IAcsClient client = new DefaultAcsClient(profile);
      FindProjectListRequest request = new FindProjectListRequest();
      //设置业务参数。
      request.setCsbId(227L);
      request.setPageNum(1);
      FindProjectListResponse response = client.getAcsResponse(request);
      System.out.println(gson.toJson(response));
    } catch (ServerException e) {
      e.printStackTrace();
    } catch (ClientException e) {
      e.printStackTrace();
    }
  }
```

示例输出结果：

```
{
 "code": 200,
 "message": "success",
 "requestId": "40FE0129-41AE-4464-9F0F-56328872623F",
 "data": {
  "currentPage": 1,
  "pageNumber": 1,
  "total": 1,
  "projectList": [
   {
    "apiNum": 11,
    "csbId": 227,
    "deleteFlag": 0,
    "description": "asdcdfdsdfddddddsasc",
    "gmtCreate": 1511164185000,
    "gmtModified": 1531134184000,
    "id": 420,
    "ownerId": "********",
    "projectName": "group2",
    "projectOwnerEmail": "group2ddd",
    "projectOwnerName": "group2ddd",
    "projectOwnerPhoneNum": "group2dddqq",
    "status": 1,
    "userId": "********"
   }
  ]
 }
}
```


Java SDK源代码下载地址。

Python SDK

Python SDK下载地址 。

Python SDK源代码下载地址。

PHP SDK

PHP SDK源代码下载地址。

FAQ

- 专有云里有些API，在公有云的管理API列表里没有？

目前只开放了用户需要的部分API，根据需要，后续可增加开放API的类别和数量。

- 管理API支持HTTP方式访问吗？

  出于安全考虑，管理API只支持HTTPS方式访问。

- 管理API提供哪些SDK?

  目前只提供了Java、Python和PHP，如果您有其它语言的需求，请您联系CSB接口人。

4.错误码

CSB提供了错误码，帮助您定位在使用CSB的SDK时遇到的问题。

错误码简介

消费端通过CSB调用服务时，整个链路上可以分成三个阶段：

1. 服务消费端应用以消费端协议访问CSB。
2. CSB处理转发服务调用请求。
3. CSB以提供端协议访问服务提供端应用。

这三个阶段都可能出错，服务消费端收到错误信息时要能区别是在哪个阶段发生了什么错误，需要考虑如何通过消费端协议定义的错误信息结构来体现。消费端协议定义的错误信息结构多种多样，也可能有各种限制，但是通常都至少会提供错误码和错误描述，而且错误码应该都有保留或者未占用的、代表“其它错误”的编码（下称：逃逸代码Escape Code）。基于这个前提，可以定义如下约定：

- 阶段1发生的错误不用做任何处理。
- 阶段2及之后发生的错误，约定使用消费端协议的某个Escape Code，具体CSB错误码和描述以固定格式体现在消费端错误描述内，如[CSB Error Code] CSB Error Message。
- 阶段3发生的错误，与阶段2相同处理，但是CSB Error Code也提供Escape Code表示这是一个阶段3错误，且CSB Error Message的格式定义为[提供端协议错误码] 提供端协议错误信息。

> **说明** CSB需要了解提供端协议的错误码中哪些表示正常访问，以免将成功的访问当做错误处理。

CSB错误码列表

其中800为CSB Escape Code 。

错误码	错误描述信息	说明
200	SUCCESS	请求处理成功
500	platform error	平台处理错误
501	access permission deny	没有权限访问
502	signature verification failed	验签失败
503	service not registed	服务没有注册
504	api name not found	该服务没有找到
505	access key not found	AccessKey没有在参数里找到
506	signature not found	签名没有在参数里面找到
507	required parameter is missing	参数丢失
508	need to access security channel	需要能访问安全通道

错误码	错误描述信息	说明
509	timestamp not found	时间戳在参数里面没有找到
510	time expired	访问过期
511	invoke timeout	调用HSF服务超时
512	write channel error	转发HSF协议时，连接通道出错。
513	connection has broken	连接已经断开
514	hsf address not found	调用时没有找到HSF服务地址
515	hsf process error	调用HSF出错
516	hoh process error	级联调用HSF出错
517	json2hsf process error	HTTP调用HSF失败
518	service metadata is null	发布的HSF服务信息为空
519	access permission deny by black list	访问被黑名单拒绝
520	can not found service metadata	调用时没有从缓存中找到HSF服务的元信息。
521	access permission deny, ip is not in white list	访问者的IP没有加入白名单
522	this invoke protocol not open	该服务没有开放成此协议
523	access permission deny due to strict accesskey and ip white list	此凭证设置了白名单，需要访问者IP加入到白名单里。
524	exceed access limition, try later	触发限流
800	service server error	服务调用出错
801	connect to service server error	连接不到服务提供者
802	serivce has offline from server	此服务已经下线
803	serivce has been stopped	服务已经在CSB上关闭了
99	hsf escap code	HSF逃逸错误码
900	Unknown code	未知错误码
1001	bad soap request	SOAP消息不正确
1002	bad response	返回结果不正确

错误码	错误描述信息	说明
1003	bad hsf subscribe info	错误的HSF订阅信息
1004	hsf invoke exception	HSF调用异常
1005	bad input soap parsing	输入SOAP消息解析失败
1006	bad output soap parsing	输出的SOAP消息解析失败
1007	bad invoke restful provider	HTTP服务Provider异常
1008	json2ws process error, bad input restufl/json request	HTTP调用WebService失败，HTTP请求不正确。
1009	json2ws process error	调用WebService服务失败
10001	process response result data filter failed	结果过滤处理结果失败

HSF错误码

错误码	错误描述信息	说明
20	OK	HSF调用成功
30	client timeout	调用端超时
31	server timeout	服务端超时
40	bad request	请求不合法
50	bad response	返回结果不合法
60	service not found	服务没有找到
70	service error	服务错误
79	connection has broken	连接已经断开
80	server error	服务提供端错误
81	Thread pool is busy	线程池繁忙
82	Communication error	通信异常
84	request is limited	请求被限流
88	server will close soon	服务很快就要关闭
90	client error	消费端错误
91	Unknown error	未知错误

消费端协议Escape Code列表

消费端协议	Escape Code
HTTP类	500
HSF	99

示例

消费端应用以HSF协议通过CSB访问一个后端HSF服务，三个阶段的错误示例如下：

阶段	描述
第一阶段错误	CSB的HSF协议处理器发现消费端发出的HSF请求格式不正确，以HSF错误40 bad request直接回复消费端。
第二阶段错误	CSB在处理消费端发出的HSF请求时，发现所访问的API并不存在，以HSF错误99 [504]所访问的服务API（taobao.unknown）在实例（instance0733）上不存在回复消费端。其中99为消费端协议HSF的Escape Code。
第三阶段错误	CSB处理消费端发出的HSF请求，在访问后端服务提供者时，HSF服务框架报错60 service not found，CSB以HSF错误99 [800][60] service not found回复消费端。其中99为消费端协议的Escape Code，800为CSB的Escape Code。
第三阶段错误	CSB处理消费端发出的HSF请求，在访问后端服务提供者时，HSF服务提供应用出错，因为HSF框架不支持返回具体应用逻辑错误信息，仅仅报错为80 server error，CSB以HSF错误99 [800][80] server error回复消费端。这里具体的应用错误只能在应用服务提供端看到，如果后端是其它比较开放的协议框架（用X代替），具体的应用错误信息可以报给CSB，这样CSB才能返回更具体的错误信息99 [800][X-Code] X-Message。