

ALIBABA CLOUD

阿里云

云服务总线 CSB
教程

文档版本：20210118

 阿里云

法律声明

阿里云提醒您在使用或阅读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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。 |
| 斜体 | 表示参数、变量。 | <code>bae log list --instanceid</code> <i>Instance_ID</i> |
| [] 或者 [a b] | 表示可选项，至多选择一个。 | <code>ipconfig [-all -t]</code> |
| { } 或者 {a b} | 表示必选项，至多选择一个。 | <code>switch {active stand}</code> |

目录

| | |
|------------------------|----|
| 1.如何在共享实例上发布和订阅服务 | 05 |
| 2.如何发布和调用SpringCloud服务 | 08 |
| 3.如何发布和调用JDBC数据服务 | 13 |
| 4.如何在线测试服务 | 17 |
| 5.如何为RAM用户授权使用CSB | 19 |
| 6.如何级联发布HTTP服务 | 23 |
| 7.如何级联发布HSF服务 | 29 |
| 8.如何使用熔断和降级保证调用方服务的可用性 | 31 |
| 9.如何管理CSB服务关联角色 | 36 |
| 10.如何使用JWT Token调用服务 | 39 |
| 11.如何使用服务生成代码调用服务 | 40 |

1.如何在共享实例上发布和订阅服务

本文介绍在共享实例上如何发布、订阅和调用服务。

部署后端服务

您可以在CSB共享实例上发布已有的后端业务服务。如果没有现成的后端业务服务，则需要先部署一个后端服务。

CSB目前支持Restful、SpringCloud、HSF、WebService、Dubbo和JDBC协议的后端服务。共享实例支持的后端服务分为VPC后端服务和公网后端服务。

- VPC后端服务

需要在VPC内部署服务和设置VPC安全组规则。

- 在VPC内部署服务：CSB共享实例只能访问实例所在区域（Region）的VPC服务，故请在当前CSB共享实例所在区域VPC的ECS上部署后端服务。
- 设置VPC安全组规则：需要配置VPC安全组规则，以便CSB共享实例的地址段（100.64.0.0/10）能访问到您VPC内的业务服务。更多信息，请参见[配置VPC网络安全组规则](#)。

- 公网后端服务

与VPC后端服务类似，只是不需要配置VPC安全组。

说明 由于CSB访问公网后端服务需要使用公网流量，涉及流量费用，故当前公网后端服务能发布到CSB共享实例上，但服务调用时无法访问。

配置VPC网络安全组规则

如果后端服务为VPC后端服务，CSB共享实例需要配置客户VPC的安全组规则，以便CSB共享实例的地址段（100.64.0.0/10）能访问到VPC内的业务服务。

VPC的安全组规则会因RESTful、SpringCloud、HSF、WebService、Dubbo和JDBC服务而有所不同。

- RESTful服务的安全组规则

配置允许CSB访问RESTful服务（如8081）的端口。

| 入方向 | 出方向 | | | | | |
|------|------|---------|----------|------------------|--------------------|----------|
| 手动添加 | 快速添加 | 全部编辑 | | | | |
| 授权策略 | 优先级 | 协议类型 | 端口范围 | 授权对象 | 描述 | 操作 |
| 允许 | 1 | 自定义 TCP | 目的: 8081 | 源: 100.64.0.0/10 | CSB共享实例访问RESTful服务 | 保存 预览 删除 |

- SpringCloud服务的安全组规则

配置CSB访问SpringCloud服务（如8086）的端口。

| 入方向 | 出方向 | | | | | |
|------|------|---------|----------|------------------|----------------------|----------|
| 手动添加 | 快速添加 | 全部编辑 | | | | |
| 授权策略 | 优先级 | 协议类型 | 端口范围 | 授权对象 | 描述 | 操作 |
| 允许 | 1 | 自定义 TCP | 目的: 8086 | 源: 100.64.0.0/10 | 共享实例访问SpringCloud、JD | 保存 预览 删除 |

- HSF服务的安全组规则

配置允许CSB访问HSF服务（如12200/12203）的端口。



• WebService服务安全组规则

配置允许CSB访问WebService服务（如9081）的端口。



• Dubbo服务的安全组规则

配置允许CSB访问ZooKeeper注册中心（如2181）和Dubbo服务（如28800）的端口。

允许CSB访问ZooKeeper注册中心的安全组规则



Dubbo服务的安全组规则



• JDBC服务的安全组规则

配置CSB访问JDBC服务（8086）的端口。




发布服务

在CSB实例上发布服务前，需要先创建服务组。更多信息，请参见[管理服务组](#)。

发布服务的具体步骤，请参见[发布后端已有服务](#)。

在共享实例中发布服务需要注意以下两点：

- 在共享实例中发布服务时需要选择对应的客户VPC。
 - 发布公网后端服务：选择非VPC服务。
 - 发布VPC后端服务：选择对应服务所在的VPC。

 **说明** 如果客户VPC列表里没有您要的VPC，请确认您的VPC所在地域是否与CSB共享实例在同一个地域。

- 在共享实例中发布服务无需审批，发布后即为激活状态。

订阅服务

如果要访问共享实例中发布的服务，需要先创建凭证，然后使用凭证订阅指定服务，由服务拥有者审批订购且通过后才能访问服务。更多信息，请参见[订阅服务](#)。

调用服务

订阅服务成功后，您可以通过CSB SDK使用命令行或代码调用订阅的服务。更多信息，请参见[调用服务](#)。

2.如何发布和调用SpringCloud服务

当您的SpringCloud服务注册到注册中心后，可以将该服务在CSB发布，供消费方调用。您也可以订购并调用在CSB中开放的其它SpringCloud服务。

前提条件


- 已创建注册中心。
- SpringCloud服务已注册到注册中心。

条件与约束

- 目前仅支持Nacos、Eureka和EDAS注册中心三种注册中心。
- 由于注册中心可能在VPC内部，csbConsole无法直接访问，故CSB控制台上无法直接展示注册中心内已有的SpringCloud服务列表。
- 目前仅支持将SpringCloud服务开放为RESTFul，不支持开放为WebService。

发布SpringCloud服务

1. 登录CSB控制台。
2. 在顶部菜单栏选择地域。
3. 在左侧导航栏单击实例列表。
4. 在实例列表页面单击具体实例名称。

 **注意** 如果您使用共享实例，请参见[共享实例信息](#)使用CSB指定的共享实例，否则会导致发布失败。共享实例仅用于体验试用，不建议正式生产使用。

5. 在实例概览页面左侧导航栏选择发布者 > 发布服务。
6. 在路由服务发布的命名服务页签中设置参数，单击下一步。



路由服务发布

当前实例: csb_aaliyun

1. 命名服务 | 2. 接入协议 | 3. 开放协议 | 4. 限制访问 | 5. 发布服务

* 服务全名

* 服务版本 1.0.0

* 所属服务组

业务日志 注: 开启业务日志将会在日志中记录请求和响应的内容, 不建议长时间开启, 以避免过大的日志量和可能的性能影响

显示更多

下一步

命名服务参数说明如下:

| 参数 | 描述 |
|------|-----------|
| 服务全名 | 要发布的服务全名。 |

| 参数 | 描述 |
|-------|--|
| 服务版本 | 服务版本。 |
| 所属服务组 | 在下拉列表中选择已创建的服务组。 |
| 业务日志 | 根据您的需要选择是否开启日志开关。 开启日志开关后，会记录业务请求和响应的内容，您可在CSB Broker的 <code>/home/admin/cloud-gateway/logs/trace.log</code> 内查看日志详情。 |

7. 在路由服务发布的接入协议页签中填写后端接入服务信息，然后单击下一步。

接入协议参数说明如下：

| 参数 | 描述 |
|-----------------------------|--|
| 路由策略 | <p>服务发布后，访问该服务的路由策略，包含直接路由和内容路由两种策略。</p> <ul style="list-style-type: none"> 直接路由：实际上是直接接入，不需要路由，只能选择一种接入协议。 内容路由：单击新增，在编辑参数设置路由条件，即定义Groovy条件规则，把接入的请求根据参数值的不同，路由到多个不同的后端接入协议或者相同的接入协议不同的接入地址。 |
| 选择一个接入协议 | 服务的接入协议，选择SpringCloud。 |
| 配置接入服务 | |
| 注册中心类型 | 要发布的服务所注册的注册中心类型。根据实际情况选择Nacos、Eureka或者EDAS注册中心。 |
| 注册中心地址（适用于Nacos和Eureka注册中心） | 注册中心的访问地址。 |
| EDAS命名空间TID（适用于EDAS注册中心） | EDAS命名空间对应的TID。可以在EDAS控制台的命名空间页面查看。 |

| 参数 | 描述 |
|--------------------|---|
| Spring Cloud服务名 | 在所选注册中心注册的SpringCloud服务。 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> ? 说明 由于注册中心可能在VPC内部，csbConsole无法直接访问，故CSB控制台上无法直接展示注册中心内已有的Spring Cloud服务列表。 </div> |
| Spring Cloud开放path | 根据实际情况设置服务开放路径。 开放路径设置为空表示匹配所有路径；支持星号(*)通配符；支持设置多个开放路径，路径之间需要换行分隔。 |
| Spring Cloud服务访问协议 | 根据实际情况选择服务访问协议，支持HTTP和HTTPS。 |
| 超时访问 | 服务的超时时间。如果需要为服务设置熔断和降级机制，则需要配置该参数。 |

其它参数可根据实际需求配置，请参见[设置接入协议](#)。

8. 在路由服务发布的开放协议页签中设置参数，单击下一步。



开放协议参数说明如下：

| 参数 | 描述 |
|--------|--|
| 服务开放类型 | 服务的开放协议类型。SpringCloud服务仅支持开放为RESTful协议。 |
| 开放Path | 服务发布后的访问Path，格式为 <code>/CSB服务版本号/CSB服务名</code> 。 |

其它参数可根据实际需求配置，请参见[设置开放协议](#)。

9. 在路由服务发布的限制访问页签中设置参数，单击下一步。限制访问的参数根据实际需求配置，请参见[设置访问限制](#)。

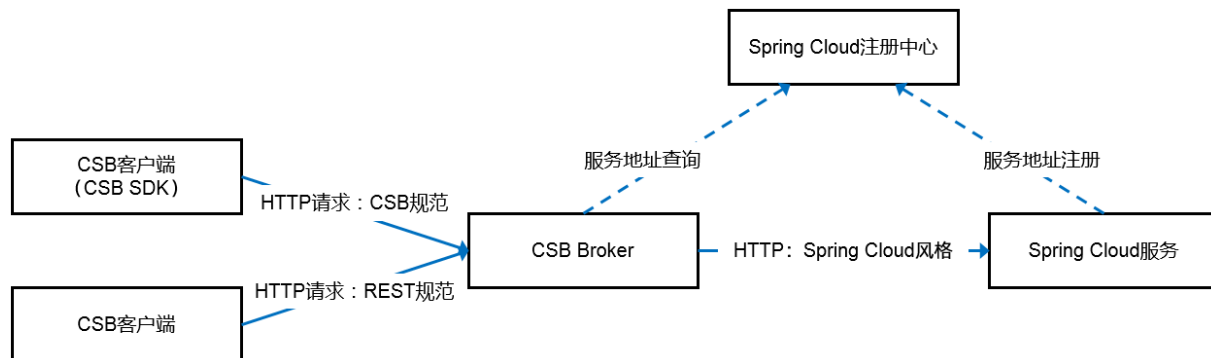
10. 在路由服务发布的发布服务页签中查看服务的基本信息、开发协议、接入协议和限制访问设置，确认无误后，单击完成发布。

订购SpringCloud服务

订购SpringCloud服务和订购其它类型的服务没有差异，请参见[订购服务](#)。

调用SpringCloud服务

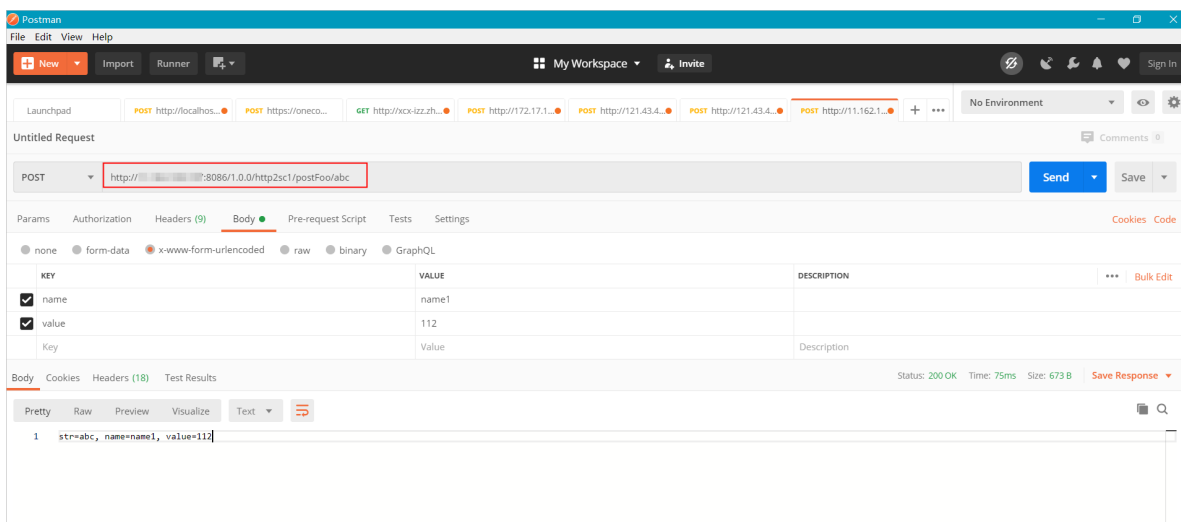
服务消费方可以通过RESTful和CSB SDK两种方式调用SpringCloud服务。



● 使用标准RESTful方式调用

? 说明 使用标准RESTful方式调用的Spring Cloud服务必须打开公开访问开关。

可以使用Postman发起调用请求，示例如下。



实例参数说明：

- `http://xx.xxx.xxx.xxx:8086/1.0.0/http2sc1`：URL格式为 `http://<CSB 服务访问IP>:<端口>/<CSB服务版本号>/<CSB服务名>`。
- `/postFoo`：SpringCloud服务要求的path路径。
- `/abc`：SpringCloud服务要求的path参数。

● 使用CSB SDK方式调用

与普通RESTful服务调用方法基本一致，请参见调用服务。包含以下区别：

- CSB服务访问地址为 `http://<csbBroker的负载均衡地址>:8086/<CSB服务版本号>/<CSB服务名> + <springCloud本身要求的path和query参数>`。
- 可以省略HttpParameters.Builder的 `api()`和 `version()`设置。

使用标准RESTful方式调用的实例如下。

```
public void testHttp2SpringCloud() {
    HttpParameters.Builder builder = new HttpParameters.Builder();
    builder.requestURL("http://xx.xxx.xxx.xxx:8086/1.0.0/http2sc1/postFoo/abc")
        .api("http2sc1") // 实例名称。
        .version("1.0.0") // 服务版本。
        .method("post") // 设置调用方式, get或post。
        .accessKey("AK值").secretKey("SK值");// 设置AccessKey ID和AccessKey Secret。
    // 设置HTTP FORM表单请求参数。
    builder.putParamsMap("name", "name1").putParamsMap("value", "123");
    try {
        String ret = HttpCaller.invoke(builder.build());
    } catch (HttpCallerException e) {
        e.printStackTrace();
    }
}
```

3.如何发布和调用JDBC数据服务

JDBC数据服务是通过JDBC将后端数据库访问能力在CSB实例上发布为服务，发布服务时只能以RESTful开放类型发布。

前提条件

- 创建或申请CSB实例，请参见[创建实例](#)。
- 了解如何在CSB实例上发布、订阅和调用RESTful服务的基本使用，请参见[快速体验开放平台](#)。
- 具有SQL和MyBatis基本使用基础知识。

背景信息

- JDBC目前支持单表和多表查询，后续根据需要会提供插入、修改和删除库表数据的能力。
- 每个CSB实例可以配置多个数据源，多个服务可以共享一个数据源。

管理数据源

CSB目前支持MySQL兼容的数据库，包括RDS Mysql、DRDS、MySQL。

CSB目前支持PostgreSQL兼容的数据库，包括RDS PostgreSQL、AnalyticDB PostgreSQL、PostgreSQL。

每个CSB实例可以配置多个数据源，多个服务可以共享一个数据源。

管理数据源的具体操作，请参见[管理数据源](#)。

发布JDBC服务

发布JDBC服务，请参见[发布后端已有服务](#)。在接入协议步骤中因单表和多表查询而配置有所不同。

- 单表查询
 - i. 在接入协议页面配置接入服务。

配置接入服务

JDBC接入协议仅支持 1.1.3.0 及以上的CSB Broker版本。

* 数据源 请选择 数据源管理

* 数据服务类别 单表查询

* 返回记录数上限 500

高级

超时时间(ms) 30000

* SQL模板 (Mybatis语法的FROM WHERE ORDER/GROUP语句)

FROM table WHERE field1 = #{参数1} and field2 in (#{参数21},#{参数22}) and field3 like concat(#{参数3},%) ORDER BY field4 asc, field5 desc

生成的SQL模板

SELECT 库表字段1 结果参数1, (表别名1.字段1 + 表别名2.字段2) * 2 复合结果字段1

生成的统计SQL模板

SELECT COUNT(*)

配置项说明：

- **数据源**：选择CSB实例上已配置好的数据源，或直接单击[数据源管理](#)进行当前CSB实例数据源的新增、修改和删除。
- **数据服务类型**：选择单表查询。

- 返回记录数上限：一次查询服务请求最多允许返回多少条记录。
- SQL模板：以MyBatis语法进行查询SQL语句模板表达，只写 FROM 表名 WHERE 条件 ORDER/GROUP 子句，CSB自动根据接入协议的出参配置进行 SELECT 结果字段列表 子句的生成。SQL模板示例如下：

```
FROM tableName
WHERE field1 > #{参数1} and field2 in (#{参数21}, #{参数22}) and field3 like concat('%',#{参数3},'%')
ORDER BY field4 asc, field5 desc
```

如果SQL模板需要包含XML转义字符时，请参见SQL模板包含XML转义字符该如何设置？。

如果SQL模板需要判断入参是否存在时，请参见SQL模板如何判断某个入参是否存在？。

- 生成的SQL模板：显示根据当前出参和SQL模板最终生成的查询SQL示例。
- 生成的统计SQL模板：显示根据当前配置和SQL模板最终生成的查询记录数统计SQL示例。
- 高级选项：COUNT扫描总记录数保护是通过设置COUNT扫描总记录数上限，对SQL语句进行封装，限定统计SQL扫描记录数量的上限，避免COUNT命令对数据库造成的压力。

ii. 在接入协议页面编辑SQL条件参数。

CSB自动生成单表查询的入参模板字段，并根据入参配置自动将请求参数传递给MyBatis语法的SQL模板，以便执行数据查询。

| 开放参数名 | SQL模板参数名 | 参数类型 | 扩展类型 | 传递方式 | 可选 | 操作 |
|-------------------|-------------------|-------------------|------|------|----|--|
| requireTotalCount | requireTotalCount | java.lang.Boolean | 正常 | | 是 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |
| offset | offset | java.lang.Integer | 正常 | | 是 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |
| limit | limit | java.lang.Integer | 正常 | | 是 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |
| queryValues | queryValues | QueryValues | 正常 | | 否 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |
| 服务入参1 | 查询参数1 | java.lang.String | 正常 | | 否 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |
| new-服务入参1 | 查询参数1 | java.lang.String | 正常 | | 否 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |
| new-new-服务入参1 | 查询参数1 | java.lang.String | 正常 | | 否 | 编辑 +添加 +添加子节点 ⌂ ⌵ ⌴ ⌶ 删除 |

各入参字段说明：

- requireTotalCount：是否返回总记录数，CSB客户端传true时，CSB会使用“统计SQL模板”进行总记录数统计。可以删除此模板字段入参，删除后不允许此服务进行总记录统计。
- offset：从第n条查询结果记录开始返回，含义与MySQL的此关键字一致。可以删除此模板字段入参，删除后不允许此服务进行offset操作。
- limit：限定本次服务请求返回的记录数，含义与MySQL的此关键字一致。可以删除此模板字段入参，删除后不允许此服务进行limit操作。
- queryValues：所有查询参数的结构体，每个子结点的“SQL模板参数”对应一个SQL模板的MyBatis参数，“发布入参名”对应客户端发起请求时传入的参数名。此结构体入参结点不可删除。

iii. 在接入协议页面编辑SQL结果参数。

CSB根据结果参数配置自动生成SQL模板的“SELECT 字段列表”子名。

| 开放参数名 | 数据表字段名 | 参数类型 | 传递方式 | 可选 | 操作 |
|------------------|---------------------------|-------------------|------|----|--|
| resultCode | resultCode | java.lang.Integer | | 否 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| resultMsg | resultMsg | java.lang.String | | 否 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| totalCount | totalCount | java.lang.Integer | | 是 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| count | count | java.lang.Integer | | 否 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| result | result | Result | | 否 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| __csbRecordError | errorMsg | java.lang.String | | 是 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| 结果参数1 | 库表字段1 | java.lang.String | | 否 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |
| 复合结果字段1 | (表别名1.字段1 + 表别名2.字段2) * 2 | java.lang.Long | | 否 | 编辑 +添加 +添加子节点 ⌕ ⌵ ⌶ ⌷ 删除 |

CSB自动生成单表查询的结果参数说明：

- resultCode：查询服务返回码，请参见**错误码**，不可删除。
 - resultMsg：查询服务执行结果消息，不可删除。
 - tatalCount：符合查询条件的总记录数，不可删除。
 - count：本次查询请求返回的记录数，不可删除。
 - result：本次查询结果记录集结构体，CSB根据此结构体子结点自动生成“SELECT 字段列表”子句，不可删除。
 - __csbRecordError：如果记录的某个字段解析错误，则此字段保存错误信息。例如出参定义某个字段为Long型，但数据表中对应字段的某条记录为字符值，则在试图解析到此记录的字段值并转换为Long型时会抛出异常，导致此条记录生成失败，此时定义的业务结果字段为空，而此字段则会表达出生成记录的错误信息。不可删除。
- 多表查询

与单表查询类似，可参见单表查询结果参数说明，只是特定配置项和字段要求略有不同。

i. 在接入协议页面配置接入服务。

配置接入服务

JDBC接入协议仅支持 1.1.3.0 及以上的CSB Broker版本。

* 数据源 数据源管理

* 数据服务类别

* 数据表名和表别名列表

* 返回记录数上限 + | -

高级

超时时间(ms)

* SQL模板 (Mybatis语法的FROM WHERE ORDER/GROUP语句)

FROM table WHERE field1 > #{参数1} and field2 in (#{参数2:1},#{参数2:2}) and field3 like concat('%',#{参数3},'%') ORDER BY field4 asc, field5 desc

生成的SQL模板

生成的统计SQL模板

配置项说明：

- 数据服务类型：选择**多表查询**。
- 数据表名和表别名列表：以英文逗号 (,) 分隔的多个数据库表和别名，如 `table1 as t1,table2 t2`。

- **SQL模板**：与单表查询不同之处在于需要明确写各表的别名，以及各字段归属表的别名，SQL模板示例如下：

```
FROM table1 alias1, table2 alias2
WHERE alias1.id=alias2.id1 and alias1.field1 > #{参数1} and alias2.field2 in (#{参数21}, #{参数22}) and
alias2.field3 like concat('%',#{参数3},'%')
ORDER BY alias1.field4 desc, alias2.field5 desc
```

- ii. 在接入协议页面编辑SQL条件参数。

多表查询与单表查询的参数配置一致，详情请参见单表查询SQL条件参数说明。

- iii. 在接入协议页面编辑SQL结果参数。

多表查询与单表查询的参数配置一致，只是需要明确字段归属的 **表别名.字段名**。

订阅JDBC服务


与订阅其它类型服务的步骤一致，请参见[订阅服务](#)。

调用JDBC服务

JDBC数据服务目前只支持RESTful协议，客户端调用时与普通RESTful服务调用方式完全一样。请参见[使用HTTP-SDK调用Console API](#)。

命令行方式调用JDBC数据服务示例如下：

```
java -jar http-sdk-1.1.5.8.jar -method post -url "http://CSB服务地址:8086/CSB" \
  -api "JDBC数据服务名" -version "服务版本号" -ak "凭证ak" -sk "凭证sk" \
  -cbJSON '{"requireTotalCount':true,'offset':跳过n条记录,'limit':返回记录数上限,'queryValues':{'查询条件字段1':'查询条件字段值1','查询条件字段2':'查询条件字段值2'}}"
```

 **说明** CSB服务地址即创建该实例时绑定的SLB的地址。

相关信息

如果您对CSB中的JDBC数据服务有任何疑问，请参见[JDBC数据服务问题](#)。


4.如何在线测试服务

CSB控制台为服务提供在线测试功能，服务的发布者和订阅者都可以在线测试服务。

功能限制

只能测试RESTful或者WebService类型的服务。

测试服务

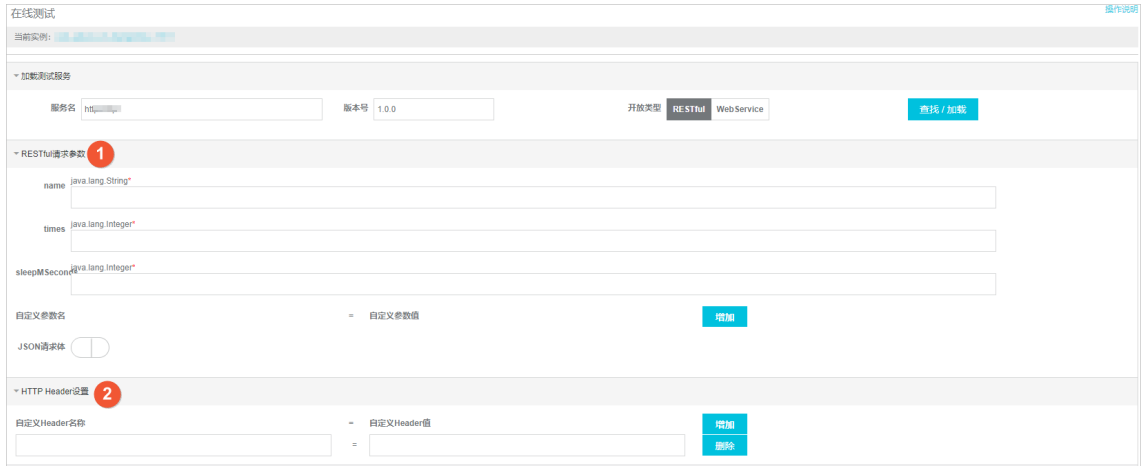
1. 登录CSB控制台。
2. 在顶部菜单栏选择地域。
3. 在左侧导航栏单击实例列表。
4. 在实例列表页面单击具体实例名称。
5. 进入测试页面。服务发布者和订阅者进入服务测试页面的路径有所不同。
 - o 发布者
 - a. 在实例概览页面左侧导航栏选择发布者 > 我的服务。
 - b. 在我的服务页面目标服务的操作列单击更多右侧的图标，在列表中单击在线测试。



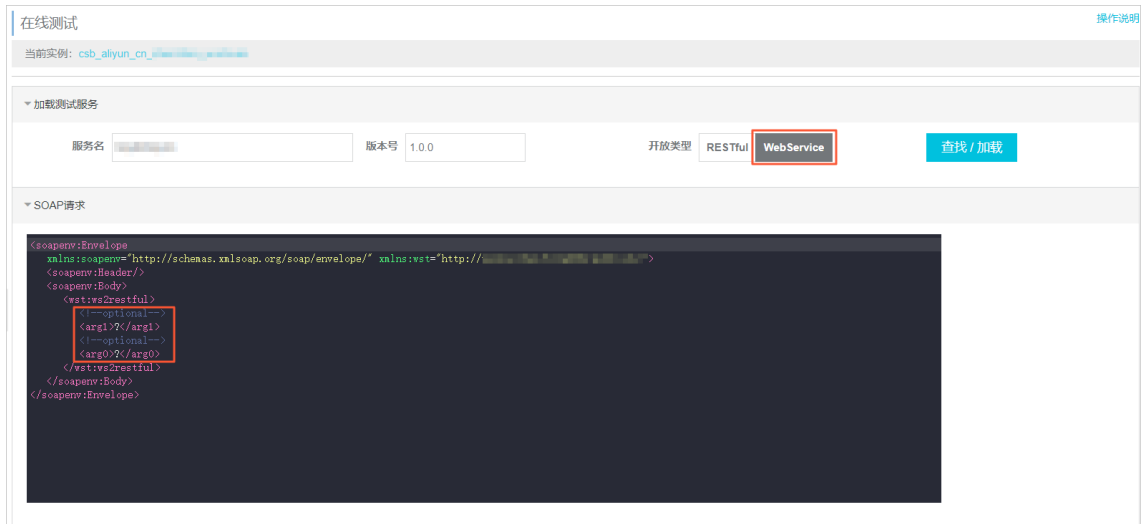
- o 订阅者
 - a. 在实例概览页面左侧导航栏选择订阅者 > 我的订购。
 - b. 在我的订购页面目标服务的操作列单击在线测试。



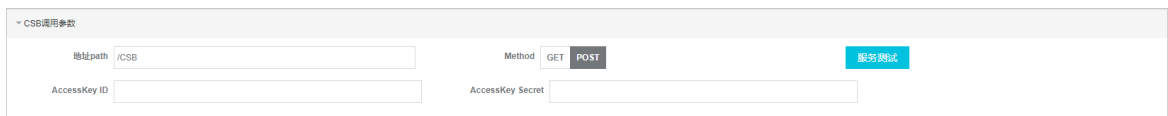
6. 在在线测试页面的加载测试服务区域选择开放类型，并在右侧单击查找/加载，加载测试请求参数。加载测试请求参数的方法会因RESTful和WebService两种开放类型而有所不同。
 - o 如果开放类型为RESTful，则根据需要，设置RESTful请求参数和HTTP Header。



- a. 根据服务定义的参数类型设置RESTful请求参数。
 - b. 如果是透传JSON Body参数，设置JSON请求体。
 - c. 根据需要设置HTTP Header，非必须设置项。
- o 如果开放类型为WebService，则会生成SOAP请求报文。根据需要，设置报文中?部分的参数值。



7. 在在线测试页面的CSB调用参数区域设置调用参数，然后单击服务测试。



CSB调用参数说明：

- o 地址path：根据实际情况填写。
- o Method：根据需要选择HTTP方法，默认值为POST。

说明 如果开放类型是WebService，Method默认为POST，不可更改。

- o AccessKey ID和AccessKey Secret：如果是订阅者测试订购的服务，则需要输入相应的AccessKey ID和AccessKey Secret。

单击服务测试后，在下面的请求报文和返回报文区域生成命令行。

5.如何为RAM用户授权使用CSB

访问控制RAM（Resource Access Management）是阿里云提供的集中式访问控制服务，其核心功能主要包括用户身份与授权管理，应用场景可以覆盖企业子账号与分权管理、针对移动App的临时授权管理，和不同组织之间的资源互操作与授权管理。本文介绍如何为RAM用户授予使用CSB的权限。

背景信息

更多关于RAM的介绍，请参见[RAM访问控制](#)。

为RAM用户授权

云账号（主账号）可以在RAM控制台创建自定义策略，然后将策略授权给RAM用户。

1. 使用云账号（主账号）登录RAM控制台创建自定义策略，请参见[创建自定义策略](#)。策略示例：
 - 全部权限示例：CsbRamAllPolicy

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

- 单个实例授权示例：CsbInstance231

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": "csb:AOSPSYS_LIST_CSB",
      "Resource": "acs:csb:*:*:AOSPSYS/*",
      "Effect": "Allow"
    },
    {
      "Action": "csb:*",
      "Resource": "acs:csb:*:*:INSTANCE/231",
      "Effect": "Allow"
    }
  ]
}
```

2. 将自定义策略授予需要的RAM用户，请参见[为RAM用户授权](#)。

RAM相关术语

为了方便您更好地使用CSB的访问控制功能，本文提供了以下几个关键术语的介绍。更多关于RAM的术语解释，请参见[基本概念](#)。

● 资源 (Resource)

资源是云服务呈现给用户与之交互的对象实体的一种抽象。CSB提供了几个类别的资源，每个资源有一个全局的阿里云资源名称 (Aliyun Resource Name, ARN)，格式为：`acs:<service-name>:<region>:<account-id>:<resource-relative-id>`。各组成部分代表的含义如下：

- acs: Alibaba Cloud Service的首字母缩写，表示阿里云的公有云平台。
- service-name: 服务名称，CSB的服务名称为csb。
- region: 区域信息，填“*”号。
- account-id: 阿里云账号ID，例如1234567890123456。
- resource-relative-id: CSB资源，`${resource-type}/${resource-id}`。

示例：名称为 `acs:csb:*:*:INSTANCE/10` 的资源表示的含义是：ID为10的CSB实例资源。

● 操作 (Action)

action规则：`csb:${action-name}`

示例：`csb:SERVICEGROUP_CREATE` 表示“创建服务组”的操作。

规则参考

● 资源规则

| 资源名 | 资源规则 |
|-----|--|
| 系统 | <code>acs:csb:\$regionid:\$accountid:AOSPSYS/*</code> |
| 实例 | <code>acs:csb:\$regionid:\$accountid:INSTANCE/\${resourceid}</code> |
| 凭证 | <code>acs:csb:\$regionid:\$accountid:CREDENTIALGROUP/\${resourceid}</code> |
| 服务组 | <code>acs:csb:\$regionid:\$accountid:SERVICEGROUP/\${resourceid}</code> |
| 服务 | <code>acs:csb:\$regionid:\$accountid:SERVICE/\${resourceid}</code> |

● 鉴权规则

| 操作 | 鉴权规则 |
|--------------------|---|
| AOSPSYS_LIST_CSB | <code>acs:csb:\$regionid:\$accountid:AOSPSYS/*</code> |
| INSTANCE_APPLY_USE | <code>acs:csb:\$regionid:\$accountid:AOSPSYS/*</code> |

| 操作 | 鉴权规则 |
|--|---|
| INSTANCE_APPROVE_APPLY_USE | acs:csb:\$regionid:\$accountid:INSTANCE/\$resourceid |
| INSTANCE_UPDATE_BASIC_INFO | acs:csb:\$regionid:\$accountid:INSTANCE/\$resourceid |
| INSTANCE_VIEW_PERSONNAL_SERVICE_STATISTICS | acs:csb:\$regionid:\$accountid:INSTANCE/\$resourceid |
| INSTANCE_VIEW_SERVICE_SUMMARY | acs:csb:\$regionid:\$accountid:INSTANCE/\$resourceid |
| CREDENTIALGROUP_CREATE | acs:csb:\$regionid:\$accountid:INSTANCE/\$resourceid |
| CREDENTIALGROUP_DELETE | acs:csb:\$regionid:\$accountid:CREDENTIALGROUP/\$resourceid |
| CREDENTIALGROUP_UPDATE | acs:csb:\$regionid:\$accountid:CREDENTIALGROUP/\$resourceid |
| SERVICEGROUP_CREATE | acs:csb:\$regionid:\$accountid:INSTANCE/\$resourceid |
| SERVICEGROUP_DELETE | acs:csb:\$regionid:\$accountid:CREDENTIALGROUP/\$resourceid |
| SERVICEGROUP_START_STOP | acs:csb:\$regionid:\$accountid:CREDENTIALGROUP/\$resourceid |
| SERVICEGROUP_UPDATE | acs:csb:\$regionid:\$accountid:CREDENTIALGROUP/\$resourceid |
| SERVICE_DELETE | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_MANAGE_ORDER | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_ORDER | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_PUBLISH | acs:csb:\$regionid:\$accountid:SERVICEGROUP/\$resourceid |
| SERVICE_UNORDER | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_UPDATE | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_VIEW_BASIC_INFO | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_VIEW_FULL_INFO | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |
| SERVICE_VIEW_ORDER_CALL_STATISTICS | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |

| 操作 | 鉴权规则 |
|---------------------------------|---|
| SERVICE_VIEW_SUMMARY_STATISTICS | acs:csb:\$regionid:\$accountid:SERVICE/\$resourceid |

6.如何级联发布HTTP服务

为了对业务服务进行统一管理，同时避免网络防火墙的繁琐配置，可以采用总分结构业务系统在CSB级联发布HTTP服务。

前提条件

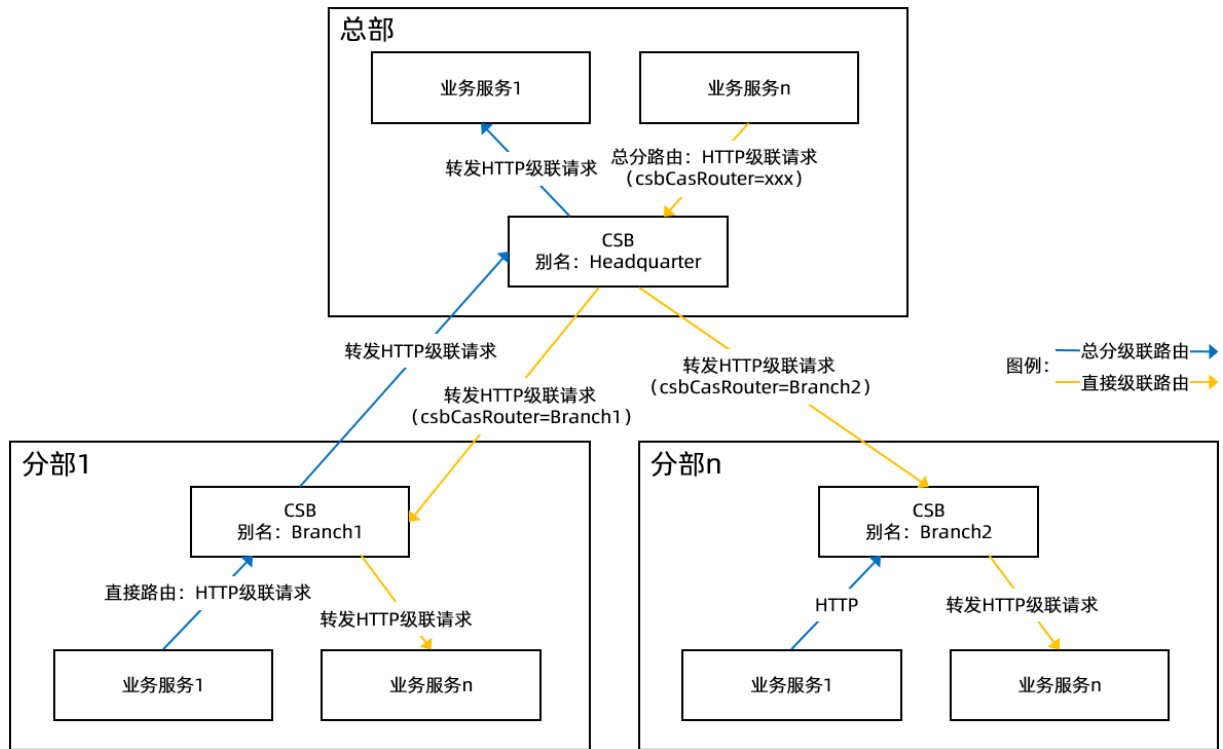
- 创建专享实例

说明 仅专享实例支持级联发布服务。

- 您需要先提交工单，由CSB技术支持人员为您配置CSB实例的级联链路。

背景信息

业务系统总分级联结构示意图如下：



说明

- 直接级联路由：接收到HTTP级联服务请求的CSB Broker直接将请求转发给指定CSB Broker级联网关地址（格式为 `http://xxx:8081`）。
- 总分级联路由：接收到HTTP级联服务请求的CSB Broker根据请求HTTP头信息中的目标CSB别名，动态路由到指定目标CSB Broker网关地址。

- 总部制定业务服务规范，各分部业务应用实现对应的服务集。也就是每个分部都会实现相同的业务服务。
- 总部与分部之间双向HTTP服务调用。

- 各分部业务服务发布到总部CSB实例，供总部应用访问（即：总部访问分部服务）：对于每个分部业务服务，总部应用希望总部CSB实例暴露一个业务服务，根据请求业务参数，由CSB自动路由到指定的分部应用。
- 总部业务服务发布到各分部CSB实例，供分部应用访问（即：分部访问总部服务）：对于总部业务服务，总部应用希望在总部CSB实例上发布一次，即可供各分部业务应用访问。

条件与约束

- 目前仅支持HTTP的总分级联。
- 只支持单附件上传和下载，不支持多附件上传。
- 总部与分部的csbConsole之间网络需要互通（端口80、443）。
- 总部与分部的csbBroker之间的网络需要互通（端口8081、8086、9081）。
- 要求总部与分部的VIP都绑定域名（如果没有统一DNS，则至少使用hosts绑定），以便后续管理。

级联发布流程

1. 创建服务组

在总部和分部的实例上创建同名的服务组。

2. 进入级联发布页面

级联发布和普通路由发布入口不同，需要进入统一级联发布页面级联发布服务。

3. 分部访问总部服务

为使分部服务访问总部服务，需要在总部CSB控制台实例上将总部服务级联发布到分部CSB实例上，且总部服务在每个分部CSB上只对应一个服务。

4. 总部访问分部服务

为使总部服务访问分部服务，需要在分部CSB控制台实例上将分部服务级联发布到总部CSB实例上。然后总部CSB实例识别多个分部来的级联服务请求，将同名的多个分部服务映射到一个级联服务。

创建服务组

进行级联发布之前，需要在总部和分部的实例上创建同名的服务组。

1. 登录CSB控制台。
2. 在顶部菜单栏选择地域。
3. 在左侧导航栏选择实例列表。
4. 在实例列表页面单击具体实例名称。
5. 在实例概览页面左侧导航栏中选择发布者 > 我的服务组。
6. 在我的服务组页面右上角单击新建服务组。
7. 在新建服务组对话框设置服务组参数，单击确认。

新建服务组
✕

* 服务组名称

* 服务组负责人

服务组负责人邮件

服务组负责人电话

服务组接口文件 上传接口包

描述信息

确认
取消

新建服务组参数说明：

| 参数 | 描述 |
|----------|--------------------------------|
| 服务组名称 | 自定义设置服务组名称。 |
| 服务组负责人 | 设置服务组的负责人。 |
| 服务组负责人邮件 | 根据实际情况设置责任人的邮件和电话，非必配项。 |
| 服务组负责人电话 | |
| 服务组接口文件 | 为这个服务组上传一个JAR类型的接口文件，用于自动生成参数。 |

新建服务组后，返回我的服务组，该服务组的状态为启动，其它信息也与配置的一致。

进入级联发布页面

在级联发布服务前，需要先进入级联发布页面。

1. 登录CSB控制台。
2. 在顶部菜单栏选择地域。
3. 在左侧导航栏单击实例列表。
4. 在实例列表页面单击具体实例名称。

注意 如果您使用共享实例，请参见[共享实例信息](#)使用CSB指定的共享实例，否则会导致发布失败。共享实例仅用于体验试用，不建议正式生产使用。

5. 在实例详情页面左侧导航栏选择发布者 > 统一级联发布。

分部访问总部服务

为使分部服务访问总部服务，总部业务人员在总部CSB控制台上选择总部CSB实例，将服务级联发布到分部的CSB实例上。

级联发布和单实例发布服务的流程基本一致，本文仅介绍级联发布服务的流程及需要注意的事项。

- 命名服务，请参见[命名服务](#)。

注意 所属服务组必须选择总部在总部CSB实例和分部CSB实例上创建的同名的服务组。

- 设置接入协议，请参见[设置接入协议](#)。
 - 路由策略：目前仅支持直接路由。
 - 接入协议：对于HTTP级联发布，则选择RESTful。
 - 端点：总部业务服务的HTTP访问地址。
 - 方法：总部业务服务要求的HTTP方法，目前支持POST和GET。
 - 请求格式：普通HTTP请求或JSON请求内容。
 - 编辑入参：可以根据业务服务的入参信息，填写入参列表。也可以不填。
 - 编辑出参：可以根据业务服务的出参信息，填写出参列表。也可以不填。
- 设置开放协议，请参见[设置开放协议](#)。
- 设置访问限制，请参见[设置访问限制](#)。

说明 如果想保证调用方服务的可用性，还可以配置熔断和降级，请参见[如何使用熔断和降级保证调用方服务的可用性](#)。

- 设置目标实例。和普通路由发布不同，级联发布还需要设置级联发布的目标实例。在目标实例页面勾选服务发布目标实例，在目标实例路由方式右侧的列表中选择总分路由，然后单击下一步。



- 在发布服务页面确认设置，单击完成发布。

总部访问分部服务

为使总部服务访问分部服务，分部业务人员在CSB控制台上选择分部CSB实例，将服务级联发布到总部CSB实例上。

- 如果总部无此级联服务，按现有CSB逻辑，直接新建总部的级联服务，并设置服务类型为直接路由服务。
- 总部已有此级联服务，增加总部已有CSB级联服务的分部服务引用关系，并设置服务类型为内容路由服务。

定义路由规则（由系统根据规则生成总部级联服务路由到分部的规则）：


- 默认以HTTP Query的csbRouter参数作为路由参数。
- 以分部CSB实例名作为csbRouter值。

级联发布和单实例发布服务的流程基本一致，本文仅介绍级联发布服务的流程及需要注意的事项。

- 命名服务，请参见[命名服务](#)。

 **注意** 所属服务组必须选择总部在总部CSB实例和分部CSB实例上创建的同名的服务组。

- 设置接入协议，请参见[设置接入协议](#)。
 - 路由策略**：目前仅支持直接路由。
 - 接入协议**：对于HTTP级联发布，则选择RESTful。
 - 端点**：总部业务服务的HTTP访问地址。
 - 方法**：总部业务服务要求的HTTP方法，目前支持POST和GET。
 - 请求格式**：普通HTTP请求或JSON请求内容。
 - 编辑入参**：可以根据业务服务的入参信息，填写入参列表。也可以不填。
 - 编辑出参**：可以根据业务服务的出参信息，填写出参列表。也可以不填。
- 设置开放协议，请参见[设置开放协议](#)。
- 设置访问限制，请参见[设置访问限制](#)。

 **说明** 如果想保证调用方服务的可用性，还可以配置熔断和降级，请参见[如何使用熔断和降级保证调用方服务的可用性](#)。

- 设置目标实例。和普通路由发布不同，级联发布还需要设置级联发布的目标实例。在目标实例页面勾选**服务发布目标实例**，在目标实例路由方式右侧的列表中选择**总分路由**，然后单击下一步。



- 在发布服务页面确认设置，单击完成发布。
-

执行结果

- 如果该实例的拥有者将服务发布审批设置为**不需要审批**，发布后状态为**激活**。
- 如果该实例的拥有者将服务发布审批设置为**一级审批**或**二级审批**，则发布后状态为**等待一次审批**或**等待二次审批**，需要实例拥有者审批后才能激活。

后续步骤

级联发布服务成功后，您可以订阅和调用服务。

- 订阅服务，请参见[订阅服务](#)。
- 调用服务因总分结构会有所不同，但都需要先下载最新版本的Java SDK，请参见[工具包下载地址](#)。
 - 分部应用调用总部服务
 - 对于CSB客户端来说，与调用普通CSB服务无异，请参见[调用服务](#)和[CSB-SDK说明](#)。

- 总部应用调用分部服务

此场景需要总部应用发起总部CSB服务请求时，携带分部CSB路由信息。以便总部CSB根据此路由信息将请求路由到正确的分部CSB实例。所以，需要总部应用在原有CSB服务调用方式基础上增加一个HttpHeader（`csbCasRouter = 分部CSB实例别名`）。更多信息，请参见[调用服务](#)。


为帮助您理解，分别列举命令行和代码两种方式的调用示例。

- 命令行调用示例

```
java -jar http-client-1.1.5.8.jar -method post -url "http://csb-broker.hq.com:8086/CSB?key1=value1" \
  -api "branch-headquarters-service1_csb_cas" -version "1.0.0" -ak "ak1" -sk "sk1" \
  -H "csbCasRouter:b01"
```

- 代码调用示例

```
public void testHttp2Http() {
    HttpParameters.Builder builder = new HttpParameters.Builder();
    builder.requestURL("http://总部CSB服务地址:8086/CSB")
        .api("总部级联服务名")
        .version("1.0.0")
        .method("post")
        .accessKey("ak值").secretKey("sk值")
        .putHeaderParamsMap("csbCasRouter","分部CSB实例别名");// 设置总部级联服务路由信息。
    // 设置HTTP FORM表单请求参数。
    builder.putParamsMap("key1", "value1");
    try {
        String ret = HttpCaller.invoke(builder.build());
        System.out.println("----- ret=" + ret);
    } catch (HttpCallerException e) {
        e.printStackTrace();
    }
}
```

 **说明** CSB总部服务地址即创建该实例时绑定的SLB的地址。

7.如何级联发布HSF服务

针对复杂的多环境、多归属互通场景，您可以跨实例级联发布服务，也就是在一个CSB实例上接入已有服务，而在另一个CSB实例上发布出来，供订阅者消费。本文介绍如何级联发布HSF服务。

前提条件

- 已经创建专享实例，请参见[创建专享实例](#)。

 说明 仅专享实例支持级联发布服务。

- 您需要先提交工单，由CSB技术支持人员为您配置CSB实例的级联链路。

背景信息

针对复杂的多环境、多归属互通场景，云服务总线提供级联发布管理机制，即跨CSB实例的服务发布，也就是在一个CSB实例上接入已有服务，而在另一个CSB实例上发布出来，供订阅者消费。

级联发布的详细介绍，请参见[级联发布](#)。

步骤一：创建服务组

进行级联发布之前，需要在两个实例上创建同名的服务组。

- 登录[CSB控制台](#)。
- 在顶部菜单栏选择地域。
- 在左侧导航栏选择实例列表。
- 在实例列表页面单击具体实例名称。
- 在实例概览页面左侧导航栏中选择发布者 > 我的服务组。
- 在我的服务组页面右上角单击新建服务组。
- 在新建服务组对话框设置服务组参数，单击确认。

新建服务组

* 服务组名称

* 服务组负责人

服务组负责人邮件

服务组负责人电话

服务组接口文件

描述信息

新建服务组参数说明：

| 参数 | 描述 |
|----------|--------------------------------|
| 服务组名称 | 自定义设置服务组名称。 |
| 服务组负责人 | 设置服务组的负责人。 |
| 服务组负责人邮件 | 根据实际情况设置责任人的邮件和电话，非必配项。 |
| 服务组负责人电话 | |
| 服务组接口文件 | 为这个服务组上传一个JAR类型的接口文件，用于自动生成参数。 |


新建服务组后，返回我的服务组，该服务组的状态为启动，其它信息也与配置的一致。

步骤二：级联发布服务


如果CSB技术支持人员已经为您创建了CSB实例的级联链路后，那么您可以在实例上发布服务。

级联发布和单实例发布服务的流程基本一致，本文仅介绍级联发布服务的流程及需要注意的事项。

1. [进入发布服务页面](#)。
2. [命名服务](#)。
3. [设置接入协议](#)。

 **说明** 级联发布无需编辑入参和出参。

4. [设置开放协议](#)。

 **说明** 配置完实例级联链路后，在开放协议页面中服务发布目标实例会有多个实例，请选择非当前发布服务的级联实例。

5. [设置访问限制](#)。
6. [发布服务](#)。

执行结果

- 如果该实例的拥有者将服务发布审批设置为不需要审批，发布后状态为激活。
- 如果该实例的拥有者将服务发布审批设置为一级审批或二级审批，则发布后状态为等待一次审批或等待二次审批，需要实例拥有者审批后才能激活。

后续步骤

级联发布服务成功后，您可以订阅和调用服务，请参见[订阅服务](#)和[调用服务](#)。

8.如何使用熔断和降级保证调用方服务的可用性

当CSB接入的目标服务（后端端点）不可用或响应过慢时，为了保证调用方（CSB客户端）整体服务的可用性，可以通过设置熔断和降级逻辑以提高响应速度，待目标服务恢复后再恢复正常调用。

背景信息

- 熔断：如果某个目标服务调用慢或者有大量调用超时，熔断该服务的调用，对后续请求进行调用降级或直接返回，快速释放资源，待目标服务情况好转再恢复调用。
- 降级：在目标服务调用超时、失败或者服务进入熔断状态时，返回默认值或调用降级服务，以避免原目标服务不可用导致服务调用失败。

条件与约束

- 熔断和降级服务需要配置**超时时间**，请参见[设置接入协议](#)。
- 熔断后尝试请求目标服务的最小间隔为5秒，查看目标服务是否已恢复。
- CSB无法区分目标服务的异常，超时和其他异常都会触发熔断。
- 开放协议和接入协议符合下表中的服务才支持降级配置。

| 开放协议 | 接入协议 | 降级默认值 | 降级端点 | 默认值格式 |
|-----------------------|------------|-------|------|--------------------|
| RESTful（普通服务和级联服务） | RESTful | 支持 | 支持 | JSON |
| | WebService | 支持 | 不支持 | JSON |
| | HSF | 支持 | 不支持 | JSON |
| | DUBBO | 支持 | 不支持 | JSON |
| RESTful | JDBC | 不支持 | 不支持 | 无 |
| WebService（普通服务和级联服务） | RESTful | 支持 | 支持 | JSON |
| | WebService | 支持 | 不支持 | XML |
| | HSF | 支持 | 不支持 | JSON |
| | DUBBO | 支持 | 不支持 | JSON |
| WebService | JDBC | 不支持 | 不支持 | 无 |
| HSF级联 | HSF | 支持 | 不支持 | 文本（只支持Hessian2序列化） |

配置熔断和降级

熔断和降级是在发布服务过程中配置的，发布服务的详细步骤，请参见[发布后端已有服务](#)，本文仅说明发布过程中如何配置熔断和降级。

1. 进入发布服务页面，请参见[进入发布服务页面](#)或[进入级联发布页面](#)。

2. 在命名服务页面设置服务基本参数，请参见命名服务。
3. 在接入协议页面选择接入协议并配置相关参数，请参见设置接入协议。

 **注意** 在配置接入服务时，必须配置超时时间，否则熔断和降级配置服务不生效。

4. 在开放协议页面设置开放协议，请参见设置开放协议。
5. 在限制访问页面设置访问限制，并进行熔断设置。设置访问限制，请参见设置访问限制，本文主要介绍熔断设置。

熔断参数配置说明：

- **启用熔断**：是否打开启用熔断开关。
- **熔断触发条件**：设置熔断触发的条件，例如 10秒内服务请求数超过 100次，且错误率大于 40%。
- **启用降级**：是否打开启用降级开关。当启用熔断不启用降级时，在熔断状态下会抛出熔断异常。
- **降级逻辑**：输入降级逻辑代码。降级逻辑包含默认值和降级端点两种方式。
 - 如果您已有备用的端点，可配置降级端点。
 - 如果您没有备用的端点，可配置默认值，以保证服务调用可以正常进行。

下面分别提供默认值和降级端点的配置示例。

- 降级端点（和RESTful端点对应，熔断状态或原端点服务异常情况下调用降级端点的服务）示例：

```
http://service.ken.com:8080/fallback/item/{name}
```

- 默认值（返回调用方指定结果）示例（RESTful）：

```
{
  "msg": "FALLBACK",
  "result": {
    "quantity": 9999
  },
  "code": "0",
  "innerMsg": "DefaultValue"
}
```

HSF或Dubbo方法返回值对应的JSON（JSON.toJSONString(result)）。

对于HSF级联发布的服务，需要使用Hessian2序列化之后的字符串作为默认值，即调用 `Hessian2Encoder.encodeToString(result)` 获得的字符串。示例代码如下（需要先在pom.xml中添加Dubbo和Hessian的依赖）：


```
<dependency>
  <groupId>org.apache.dubbo</groupId>
  <artifactId>dubbo</artifactId>
  <version>2.7.5</version>
</dependency>
<dependency>
  <groupId>com.taobao.hsf.hessian</groupId>
  <artifactId>hessian</artifactId>
  <version>4.0.7.bugfix12-tuning3</version>
</dependency>
```

```
import org.apache.dubbo.common.io.UnsafeByteArrayInputStream;
import org.apache.dubbo.common.io.UnsafeByteArrayOutputStream;
import com.taobao.hsf.com.caucho.hessian.io.Hessian2Input;
import com.taobao.hsf.com.caucho.hessian.io.Hessian2Output;
import com.taobao.hsf.com.caucho.hessian.io.SerializerFactory;
public class Hessian2Encoder {
    private static final SerializerFactory factory = new com.taobao.hsf.com.caucho.hessian.io.Seriali
zerFactory();
    private static final String chars = "0123456789ABCDEF";
    public static byte[] encode(Object object) throws Exception {
        UnsafeByteArrayOutputStream byteArray = new UnsafeByteArrayOutputStream();
        Hessian2Output output = new Hessian2Output(byteArray);
        output.setSerializerFactory(factory);
        output.writeObject(object);
        output.close();
        byte[] bytes = byteArray.toByteArray();
        return bytes;
    }
    public static String encodeToString(Object object) throws Exception {
        byte[] bytes = encode(object);
        return bytesToHexString(bytes);
    }
    private static String bytesToHexString(byte[] bytes) {
        StringBuilder result = new StringBuilder(bytes.length);
        String s;
        for (int i = 0; i < bytes.length; i++) {
            s = Integer.toHexString(0xFF & bytes[i]);
            if (s.length() < 2) {
                result.append(0);
            }
            result.append(s.toUpperCase());
        }
        return result.toString();
    }
}
```

6. 在发布服务页面确认设置，单击完成发布。

结果验证

熔断和降级配置完成后，在调用该服务时，返回的HTTP Response包含以下Header信息，您可以根据各参数的返回值判断熔断和降级是否生效。

HTTP Response中Header信息的参数说明如下：

- `Response-Source` : Header的Key
EndPoint（源端点响应）/FallbackDefaultValue（默认值）/FallbackEndPoint（降级端点响应）
- `SuccessfulExecution` : 执行是否成功
- `ResponseRejected` : 是否拒绝
- `ResponseTimedOut` : 是否超时
- `ResponseShortCircuited` : 是否熔断（断路）
- `ResponseFromFallback` : 是否降级响应

9.如何管理CSB服务关联角色

本文介绍CSB服务关联角色（AliyunServiceRoleForCSB）的应用场景以及如何删除服务关联角色。

背景信息

CSB服务关联角色（AliyunServiceRoleForCSB）是为了完成CSB自身的某个功能，需要获取其他云服务的访问权限，而提供的RAM角色。更多关于服务关联角色的信息，请参见[服务关联角色](#)。

应用场景

CSB的创建Broker运行节点功能需要访问VPC、ECS、SLB云服务的资源，通过服务关联角色功能获取访问权限。

AliyunServiceRoleForCSB的权限策略

AliyunServiceRoleForCSB的角色权限策略为AliyunServiceRolePolicyForCSB。权限说明如下：

```
{
  "Action": [
    "vpc:DescribeVpcs",
    "vpc:DescribeVSwitches",
    "vpc:DescribeRegions",
    "vpc:DescribeZones",
    "ecs:DescribeInstances",
    "ecs:DescribeInstanceStatus",
    "ecs:DescribeRegions",
    "ecs:DescribeZones",
    "ecs:CreateNetworkInterface",
    "ecs:DeleteNetworkInterface",
    "ecs:DescribeNetworkInterfaces",
    "ecs:CreateNetworkInterfacePermission",
    "ecs:DescribeNetworkInterfacePermissions",
    "ecs:DescribeSecurityGroups",
    "ecs:DescribeSecurityGroupAttribute",
    "ecs:DescribeSecurityGroupReferences",
    "slb:AddBackendServers",
    "slb:RemoveBackendServers",
    "slb:CreateLoadBalancerTCPLListener",
    "slb:DescribeLoadBalancerTCPLListenerAttribute",
    "slb:SetLoadBalancerTCPLListenerAttribute",
    "slb:CreateLoadBalancerHTTPListener",
    "slb:DescribeLoadBalancerHTTPListenerAttribute",
    "slb:SetLoadBalancerHTTPListenerAttribute",
    "slb:CreateLoadBalancerHTTPSListener",
    "slb:DescribeLoadBalancerHTTPSListenerAttribute",
    "slb:SetLoadBalancerHTTPSListenerAttribute",
    "slb>DeleteLoadBalancerListener",
    "slb:DescribeLoadBalancers",
    "slb:DescribeLoadBalancerAttribute",
    "slb:DescribeHealthStatus",
    "slb:DescribeLoadBalancers",
    "slb:DescribeLoadBalancerAttribute",
    "slb:DescribeHealthStatus"
  ],
  "Resource": "*",
  "Effect": "Allow"
}
```

删除服务关联角色

如果您需要删除AliyunServiceRoleForCSB（服务关联角色），需要先暂停或删除依赖这个服务关联角色的CSB实例（共享实例除外）。删除实例的具体操作，请参见[暂停和删除实例](#)。

1. 云账号登录[RAM控制台](#)。
2. 在左侧导航栏，单击[RAM角色管理](#)。
3. 在[RAM角色管理](#)页面的搜索框中输入*AliyunServiceRoleForCSB*。
4. 在[RAM角色名称](#)列表中该角色的操作列单击删除。
5. 在[删除RAM角色](#)对话框单击确定。

10.如何使用JWT Token调用服务

CSB开发平台支持创建JWT类型的凭证，API消费方可直接使用JWT Token调用订阅的服务。


背景信息

CSB目前支持SDK调用服务和JWT Token调用服务，如果您想要了解SDK调用服务，请参见[调用服务](#)。

准备工作

在调用服务前，请确保您已经完成以下工作：

- [发布服务](#)。
- [创建JWT凭证](#)。
- [订阅服务](#)。
- 获取CSB上发布的服务的访问地址。
 - 获取共享实例的服务地址，请参见[共享实例名称和服务访问地址](#)。
 - 专享实例的服务地址格式为 `http://CSB服务地址:8086/CSB`。

 **说明** CSB服务地址即创建该实例时绑定的SLB的地址。

CSB服务地址的获取方式如下：

- 使用创建该实例的账号登录[EDAS控制台](#)。
- 在左侧导航栏选择组件中心 > 云服务总线。
- 在云服务总线左侧导航栏单击实例管理。
- 在实例列表页面选择目标地域，然后查看并获取目标实例的SLB地址。



| 实例名称 | 运行状态 | VPC | 交换机 | 安全组 | SLB | SLB地址 | SLB状态 | 操作 |
|--------------------------------|------|-----|-----|-----|-----|-------------------------|-------|---|
| csb_siyun_cn_shanghai_share001 | 运行 | | | | | 无 | 无 | 查看详情 更多 |
| csb_siyun_cn_shanghai | 运行 | | | | | 查看SLB地址 | 运行 | 查看详情 更多 |

调用服务

使用JWT Token调用服务的方式如下：

```
curl -H "Authorization:Bearer $jwtToken" -d "参数kv值" http://CSB服务地址:8086/CSB服务版本号/CSB服务名
```

说明

- jwtToken即是我的凭证页面目标凭证的当前凭证列下的jwt值，jwt值携带有效期字段，请删除后使用。
- CSB服务地址即创建该实例时绑定的SLB的地址。

11.如何使用服务生成代码调用服务

已经发布到CSB开放平台的服务，能自行生成Java代码。API消费方可直接使用Java代码调用订阅的服务。

背景信息

除本文介绍的使用Java代码调用服务外，CSB目前还支持SDK调用服务和JWT Token调用服务。如果您想要了解更多调用服务的方式，请参见：

- [使用SDK调用服务](#)
- [使用JWT Token调用服务](#)

准备工作

在调用服务前，请确保您已经完成以下工作：

- [发布服务](#)
- [创建凭证](#)
- [订阅服务](#)

获取服务访问地址

服务发布到共享实例和专享实例，服务访问地址是不一样的。

- 获取共享实例的服务地址，请参见[共享实例名称和服务访问地址](#)。
- 专享实例的服务地址格式为 `http://CSB服务地址:8086/CSB`。

? 说明 CSB服务地址即创建CSB实例时绑定的SLB的地址。

CSB服务地址的获取方式如下：

- 使用创建CSB实例的账号登录[EDAS控制台](#)。
- 在左侧导航栏选择[组件中心 > 云服务总线](#)。
- 在云服务总线左侧导航栏单击[实例管理](#)。
- 在实例列表页面选择目标地域，然后查看并获取目标实例的SLB地址。

| 实例名称 | 运行状态 | VPC | 交换机 | 安全组 | SLB | SLB地址 | SLB状态 | 操作 |
|----------------------------------|------|-----|-----|-----|-----|---|-------|----|
| csb_shiyun_cn_shanghai_shared001 | 运行 | | | | | 无 | 无 | 更多 |
| csb_shiyun_cn_shanghai | 运行 | | | | | 无 | 运行 | 更多 |

获取服务生成代码

1. 登录[CSB控制台](#)。
2. 在顶部菜单栏选择[地域](#)。
3. 在左侧导航栏单击[实例列表](#)。
4. 在实例列表页面单击具体实例名称。
5. 在左侧导航栏选择[发布者 > 我的服务](#)。
6. 在我的服务页面选择目标服务，在该服务的操作列下选择[更多 > 生成代码](#)。
7. 在生成调用代码页面拷贝并记录CSB-SDK依赖和Java代码。

调用服务

本示例以在IntelliJ IDEA创建工程，使用服务生成代码调用服务为例。

1. 打开IntelliJ IDEA，并创建一个工程。
2. 在文件中添加保存的CSB-SDK依赖。

```
<dependency>
  <groupId>com.alibaba.csb.sdk</groupId>
  <artifactId>http-client</artifactId>
  <version>1.1.5.8</version>
</dependency>
```

3. 在应用中添加保存的Java代码，并设置入参的值。本示例以发布的RESTful服务生成的Java代码为例：

```
public static void main(String[] args) {
    Map<String, Object> params = new HashMap<>();
    params.put("name", "name");
    params.put("times", 0);
    params.put("sleepMSeconds", 0);
    HttpParameters.Builder builder = HttpParameters.newBuilder()
        .requestURL("http://192.168.0.163:8086/CSB")
        .api("http2http1")
        .version("1.0.0")
        .method("POST")
        .contentBody(new ContentBody(JSON.toJSONString(params)));
    HttpReturn response = null;
    try {
        response = HttpClient.invokeReturn(builder.build());
    } catch (HttpClientException e) {
        throw new RuntimeException(e);
    }
    if (response.httpCode == 200) {
        System.out.println("调用CSB服务返回:" + response.getResponseStr());
        String result = response.getResponseStr();
    }
}
```

4. 运行程序，调用服务。调用服务返回的结果示例如下：

```
200 OK|
调用CSB服务返回:{"result":"JSON result is: Hello null love null times. jsonbody={"times":12,"name":"name1","sleepMSeconds":0}"}
```