

企业路由器

## API 参考

文档版本 01  
发布日期 2025-03-20



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

# 目 录

- 1 使用前必读..... 1
  - 1.1 概述..... 1
  - 1.2 调用说明..... 1
  - 1.3 终端节点（Endpoint）..... 1
  - 1.4 约束与限制..... 1
  - 1.5 基本概念..... 2
- 2 API 概览..... 3
- 3 如何调用 API..... 4
  - 3.1 构造请求..... 4
  - 3.2 认证鉴权..... 7
  - 3.3 返回结果..... 9
- 4 API..... 11
  - 4.1 企业路由器..... 11
    - 4.1.1 创建企业路由器..... 11
    - 4.1.2 更新企业路由器..... 20
    - 4.1.3 查询企业路由器详情..... 27
    - 4.1.4 查询企业路由器列表..... 32
    - 4.1.5 更新企业路由器的可用区信息..... 39
    - 4.1.6 删除企业路由器..... 45
  - 4.2 VPC 连接..... 48
    - 4.2.1 创建 VPC 连接..... 48
    - 4.2.2 更新 VPC 连接基本信息..... 56
    - 4.2.3 查询 VPC 连接详情..... 62
    - 4.2.4 查询 VPC 连接列表..... 67
    - 4.2.5 删除 VPC 连接..... 73
  - 4.3 其他连接..... 77
    - 4.3.1 更新连接基本信息..... 77
    - 4.3.2 查询连接详情..... 83
    - 4.3.3 查询连接列表..... 88
    - 4.3.4 接受连接创建..... 95
    - 4.3.5 拒绝连接创建..... 101
  - 4.4 路由表..... 106

|                       |            |
|-----------------------|------------|
| 4.4.1 创建路由表.....      | 106        |
| 4.4.2 更新路由表信息.....    | 114        |
| 4.4.3 查询路由表详情.....    | 120        |
| 4.4.4 查询路由表列表.....    | 124        |
| 4.4.5 删除路由表.....      | 130        |
| 4.5 关联.....           | 134        |
| 4.5.1 创建路由关联.....     | 134        |
| 4.5.2 查询路由关联列表.....   | 140        |
| 4.5.3 删除路由关联.....     | 146        |
| 4.6 传播.....           | 150        |
| 4.6.1 创建路由传播.....     | 150        |
| 4.6.2 查询路由传播列表.....   | 156        |
| 4.6.3 删除路由传播.....     | 162        |
| 4.7 路由.....           | 166        |
| 4.7.1 创建静态路由.....     | 166        |
| 4.7.2 更新静态路由.....     | 172        |
| 4.7.3 查询静态路由详情.....   | 178        |
| 4.7.4 查询静态路由列表.....   | 182        |
| 4.7.5 查询有效路由列表.....   | 188        |
| 4.7.6 删除静态路由.....     | 194        |
| 4.8 标签.....           | 197        |
| 4.8.1 查询项目标签.....     | 197        |
| 4.8.2 查询资源标签.....     | 199        |
| 4.8.3 创建资源标签.....     | 204        |
| 4.8.4 批量添加删除资源标签..... | 210        |
| 4.8.5 删除资源标签.....     | 218        |
| 4.9 配额管理.....         | 222        |
| 4.9.1 查询配额.....       | 222        |
| 4.10 可用区.....         | 227        |
| 4.10.1 查询可用区列表.....   | 228        |
| 4.11 流日志.....         | 231        |
| 4.11.1 创建流日志.....     | 232        |
| 4.11.2 查询流日志列表.....   | 238        |
| 4.11.3 查询流日志详情.....   | 241        |
| 4.11.4 更新流日志基本信息..... | 245        |
| 4.11.5 删除流日志.....     | 250        |
| 4.11.6 开启流日志.....     | 254        |
| 4.11.7 关闭流日志.....     | 258        |
| <b>5 权限和授权项.....</b>  | <b>263</b> |
| 5.1 权限及授权项说明.....     | 263        |
| 5.2 基于角色授权的服务授权项..... | 263        |
| 5.3 企业路由器.....        | 264        |

|                  |            |
|------------------|------------|
| 5.4 VPC 连接.....  | 265        |
| 5.5 其他连接.....    | 266        |
| 5.6 路由表.....     | 266        |
| 5.7 关联.....      | 267        |
| 5.8 传播.....      | 268        |
| 5.9 路由.....      | 268        |
| 5.10 流日志.....    | 269        |
| 5.11 标签.....     | 270        |
| 5.12 配额管理.....   | 270        |
| <b>A 附录.....</b> | <b>271</b> |
| A.1 状态码.....     | 271        |
| A.2 错误码.....     | 272        |
| A.3 获取项目 ID..... | 276        |

# 1 使用前必读

## 1.1 概述

欢迎使用企业路由器（Enterprise Router, ER）。企业路由器可以连接虚拟私有云（Virtual Private Cloud, VPC）或本地网络来构建中心辐射型组网，是云上大规格，高带宽，高性能的集中路由器。企业路由器使用边界网关协议（Border Gateway Protocol, BGP），支持路由学习、动态选路以及链路切换，极大的提升网络的可扩展性及运维效率，从而保证业务的连续性。

您可以使用本文档提供API对企业路由器进行相关操作，如创建企业路由器、在企业路由器中添加连接、创建路由表、添加路由等。支持的全部操作请参见[API概览](#)。

在调用企业路由器API之前，请确保已经充分了解企业路由器相关概念，详细信息请参见“[什么是企业路由器](#)”。

## 1.2 调用说明

企业路由器提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用。

调用方法请参见[如何调用API](#)。

## 1.3 终端节点（Endpoint）

终端节点（Endpoint）即调用API的[请求地址](#)，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询服务的终端节点。

## 1.4 约束与限制

您能创建的企业路由器资源的数量与配额有关系，如果您想查看服务配额、扩大配额，具体请参见“[申请扩大配额](#)”。

更详细的限制请参见具体API的说明。

## 1.5 基本概念

- 账号

用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用用户进行日常管理工作。

- 用户

由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

- 区域（Region）

从地理位置和网络时延维度划分，同一个Region内共享弹性计算、块存储、对象存储、VPC网络、弹性公网IP、镜像等公共服务。Region分为通用Region和专属Region，通用Region指面向公共租户提供通用云服务的Region；专属Region指只承载同一类业务或只面向特定租户提供业务服务的专用Region。

详情请参见[区域和可用区](#)。

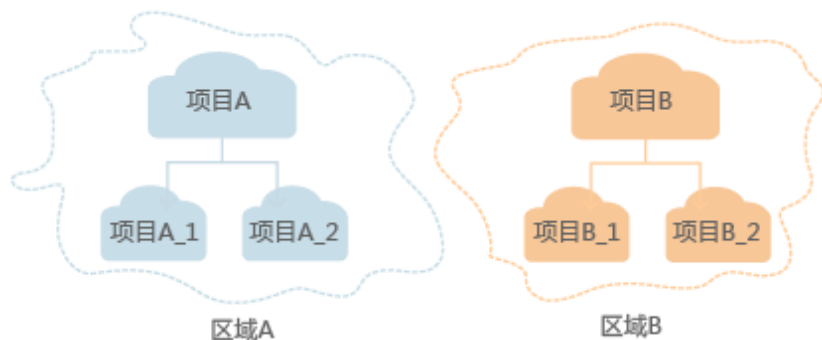
- 可用区（AZ，Availability Zone）

一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

- 项目

区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中的资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



- 企业项目

企业项目是项目的升级版，针对企业不同项目间的资源进行分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。

关于企业项目ID的获取及企业项目特性的详细信息，请参见《[企业管理用户指南](#)》。

# 2 API 概览

通过使用企业路由器服务所提供的接口，您可以完整地使用企业路由器服务的所有功能。

表 2-1 接口说明

| 类型    | 说明                                                                |
|-------|-------------------------------------------------------------------|
| 企业路由器 | 您可以通过企业路由器相关的API，执行查询企业路由器实例列表、创建企业路由器实例、更新企业路由器实例以及删除企业路由器实例等操作。 |
| VPC连接 | 您可以通过VPC连接相关的API，执行查询VPC连接列表、创建VPC连接、更新VPC连接以及切换VPC连接的子网等操作。      |
| 其他连接  | 您可以通过连接相关的API，针对VGW连接、Peering连接，执行更新连接信息、查询连接详情以及查询连接列表等操作。       |
| 路由表   | 您可以通过路由表相关的API，执行查询路由表列表、创建路由表、更新路由表以及删除路由表等操作。                   |
| 关联    | 您可以通过关联相关的API，执行查询路由表关联列表、在路由表创建以及删除关联等操作。                        |
| 传播    | 您可以通过传播相关的API，执行查询路由传播列表、在路由表创建以及删除传播等操作。                         |
| 路由    | 您可以通过路由相关的API，执行查询路由、修改路由、创建及删除静态路由等操作。                           |
| 标签    | 您可以通过标签相关的API，执行通过标签查询项目标签、查询资源标签、创建以及删除资源标签等操作。                  |
| 配额    | 您可以通过配额相关的API，执行查询企业路由器使用量、VPC连接使用量等操作。                           |
| 可用区   | 您可以通过可用区API，执行查询支持创建企业路由器实例的可用区列表操作。                              |
| 流日志   | 您可以通过流日志相关的API，执行创建流日志、查询流日志、开启以及关闭流日志等操作。                        |

# 3 如何调用 API

## 3.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的[管理员创建IAM用户](#)来说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

### 请求 URI

请求URI由如下部分组成：

**{URI-scheme}://{Endpoint}/{resource-path}?{query-string}**

尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

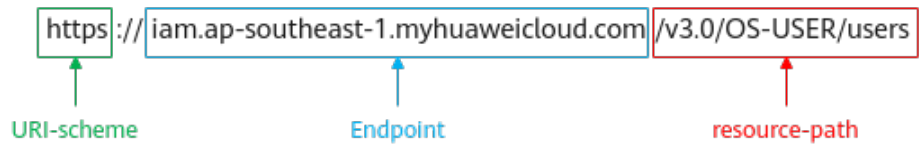
表 3-1 URI 中的参数说明

| 参数            | 描述                                                                                                                                               |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| URI-scheme    | 表示用于传输请求的协议，当前所有API均采用 <b>HTTPS</b> 协议。                                                                                                          |
| Endpoint      | 指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从 <a href="#">地区和终端节点</a> 获取。<br>例如IAM服务在“中国-香港”区域的Endpoint为“iam.ap-southeast-1.myhuaweicloud.com”。 |
| resource-path | 资源路径，即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。                                                                   |
| query-string  | 查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“？limit=10”，表示查询不超过10条数据。                                                               |

例如您需要创建IAM用户，由于IAM为全局服务，则使用任一区域的Endpoint，比如“中国-香港”区域的Endpoint（iam.ap-southeast-1.myhuaweicloud.com），并在[管理员创建IAM用户](#)的URI部分找到resource-path（/v3.0/OS-USER/users），拼接起来如下所示。

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

图 3-1 URI 示意图



说明

为方便查看，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务您正在请求什么类型的操作。

表 3-2 HTTP 方法

| 方法     | 说明                                              |
|--------|-------------------------------------------------|
| GET    | 请求服务器返回指定资源。                                    |
| PUT    | 请求服务器更新指定资源。                                    |
| POST   | 请求服务器新增资源或执行特殊操作。                               |
| DELETE | 请求服务器删除指定资源，如删除对象等。                             |
| HEAD   | 请求服务器资源头部。                                      |
| PATCH  | 请求服务器更新资源的部分内容。<br>当资源不存在的时候，PATCH可能会去创建一个新的资源。 |

在[管理员创建IAM用户](#)的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

详细的公共请求消息头字段请参见[表3-3](#)。

表 3-3 公共请求消息头

| 名称             | 描述                                                                                                                                  | 是否必选                                                             | 示例                                                                  |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|---------------------------------------------------------------------|
| Host           | 请求的服务器信息，从服务API的URL中获取。值为hostname[:port]。端口缺省时使用默认的端口，https的默认端口为443。                                                               | 否<br>使用AK/SK认证时该字段必选。                                            | code.test.com<br>or<br>code.test.com:443                            |
| Content-Type   | 消息体的类型（格式）。推荐用户使用默认值application/json，有其他取值时会在具体接口中专门说明。                                                                             | 是                                                                | application/json                                                    |
| Content-Length | 请求body长度，单位为Byte。                                                                                                                   | 否                                                                | 3495                                                                |
| X-Project-Id   | project id，项目编号。请参考 <a href="#">获取项目ID</a> 章节获取项目编号。                                                                                | 否<br>如果是专属云场景采用AK/SK认证方式的接口请求，或者多project场景采用AK/SK认证的接口请求，则该字段必选。 | e9993fc787d94b6c886cb<br>aa340f9c0f4                                |
| X-Auth-Token   | 用户Token。<br>用户Token也就是调用 <a href="#">获取用户Token</a> 接口的响应值，该接口是唯一不需要认证的接口。<br>请求响应成功后在响应消息头（Headers）中包含的“X-Subject-Token”的值即为Token值。 | 否<br>使用Token认证时该字段必选。                                            | 注：以下仅为Token示例片段。<br>MIIPAgYJKoZlhvcNAQcCo<br>...ggg1BBIIlNPXsidG9rZ |

## 说明

API同时支持使用AK/SK认证，AK/SK认证使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

AK/SK认证的详细说明请参见[认证鉴权](#)的“AK/SK认证”。

对于[管理员创建IAM用户](#)接口，使用AK/SK方式认证时，添加消息头后的请求如下所示。

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****
```

## 请求消息体（可选）

该部分可选。请求消息体通常以结构化格式（如JSON或XML）发出，与请求消息头中Content-Type对应，传递除请求消息头之外的内容。若请求消息体中的参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于[管理员创建IAM用户](#)接口，您可以从接口的请求部分看到所需的请求参数及参数说明，将消息体加入后的请求如下所示，其中加粗的字段需要根据实际值填写。

- **accountid**为IAM用户所属的账号ID。
- **username**为要创建的IAM用户名。
- **email**为IAM用户的邮箱。
- **\*\*\*\*\***为IAM用户的登录密码。

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3.0/OS-USER/users
Content-Type: application/json
X-Sdk-Date: 20240416T095341Z
Authorization: SDK-HMAC-SHA256 Access=*****, SignedHeaders=content-type;host;x-sdk-date,
Signature=*****

{
  "user": {
    "domain_id": "accountid",
    "name": "username",
    "password": "*****",
    "email": "email",
    "description": "IAM User Description"
  }
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用[curl](#)、[Postman](#)或直接编写代码等方式发送请求调用API。对于获取用户Token接口，返回的响应消息头中的“X-Subject-Token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

## 3.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证调用请求。

- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。推荐使用AK/SK认证，其安全性比Token认证要高。

## Token 认证

### 说明

Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头中，从而通过身份认证，获得操作API的权限。Token可通过调用[获取用户Token](#)接口获取。

调用本服务API需要项目级别的Token，即调用[获取用户Token](#)接口时，请求body中**auth.scope**的取值需要选择**project**，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", //IAM用户名
          "password": "*****", //IAM用户密码
          "domain": {
            "name": "domainname" //IAM用户所属账号名
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx" //项目名称
      }
    }
  }
}
```

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

```
POST https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

## AK/SK 认证

### 说明

AK/SK签名认证方式仅支持消息体大小在12MB以内，12MB以上的请求请使用Token认证。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK（Access Key ID）：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK（Secret Access Key）：私有访问密钥。与访问密钥ID结合使用，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名，也可以使用专门的签名SDK对请求进行签名。详细的签名方法和SDK使用方法请参见[API签名指南](#)。

#### 说明

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

## 3.3 返回结果

### 状态码

请求发送以后，您会收到响应，其中包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于[管理员创建IAM用户](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

### 响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于[管理员创建IAM用户](#)接口，返回如[图3-2](#)所示的消息头，其中“X-Subject-Token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

#### 说明

建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。

图 3-2 管理员创建 IAM 用户响应消息头

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fcec8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_ null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

### 响应消息体（可选）

该部分可选。响应消息体通常以结构化格式（如JSON或XML）返回，与响应消息头中Content-Type对应，传递除响应消息头之外的内容。

对于[管理员创建IAM用户](#)接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "user": {
```

```
{  "id": "c131886aec...",
  "name": "IAMUser",
  "description": "IAM User Description",
  "areacode": "",
  "phone": "",
  "email": "****@***.com",
  "status": null,
  "enabled": true,
  "pwd_status": false,
  "access_mode": "default",
  "is_domain_owner": false,
  "xuser_id": "",
  "xuser_type": "",
  "password_expires_at": null,
  "create_time": "2024-05-21T09:03:41.000000",
  "domain_id": "d78cbac1.....",
  "xdomain_id": "30086000.....",
  "xdomain_type": "",
  "default_project_id": null
}
```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{  "error_msg": "The request message format is invalid.",
  "error_code": "IMG.0001"
}
```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

# 4 API

## 4.1 企业路由器

### 4.1.1 创建企业路由器

#### 功能介绍

创建企业路由器实例，如果使能默认关联路由表或使能默认传递路由表，那么系统会默认创建一张路由表，作为默认关联路由表或默认传递路由表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/enterprise-router/instances

表 4-1 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

## 请求参数

表 4-2 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-3 请求 Body 参数

| 参数       | 是否必选 | 参数类型                          | 描述    |
|----------|------|-------------------------------|-------|
| instance | 是    | CreateEnterpriseRouter object | 企业路由器 |

表 4-4 CreateEnterpriseRouter

| 参数                         | 是否必选 | 参数类型                 | 描述                                                                        |
|----------------------------|------|----------------------|---------------------------------------------------------------------------|
| name                       | 是    | String               | 企业路由器实例名称，取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-（中划线）、.（点）<br>最小长度：1<br>最大长度：64 |
| description                | 否    | String               | 企业路由器实例描述信息<br>最小长度：0<br>最大长度：255                                         |
| asn                        | 是    | Long                 | 企业路由器实例的BGP AS号                                                           |
| enterprise_project_id      | 否    | String               | 企业项目ID                                                                    |
| charge_mode                | 否    | String               | 计费模式 按需<br>缺省值：postPaid                                                   |
| tags                       | 否    | Array of Tag objects | 标签信息                                                                      |
| enable_default_propagation | 否    | Boolean              | 是否开启默认路由表传播，默认false不开启<br>缺省值：false                                       |

| 参数                             | 是否必选 | 参数类型             | 描述                                                  |
|--------------------------------|------|------------------|-----------------------------------------------------|
| enable_default_association     | 否    | Boolean          | 是否开启默认路由表关联，默认false不开启<br>缺省值： <b>false</b>         |
| availability_zone_ids          | 是    | Array of strings | 企业路由器所在的可用区列表<br>最小长度： <b>0</b><br>最大长度： <b>128</b> |
| auto_accept_shared_attachments | 否    | Boolean          | 是否开启自动接受共享连接创建，默认false不开启<br>缺省值： <b>false</b>      |

表 4-5 Tag

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                  |
|-------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | 否    | String | 标签键，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>_sys_开头属于系统标签，租户不能输入。</li><li>标签key不能为空（长度不能为0）。</li><li>长度在1-128之间。</li></ul> 最小长度： <b>1</b><br>最大长度： <b>128</b> |
| value | 否    | String | 标签值，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> 最小长度： <b>0</b><br>最大长度： <b>255</b>                        |

## 响应参数

状态码：202

表 4-6 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-7 响应 Body 参数

| 参数         | 参数类型                                    | 描述    |
|------------|-----------------------------------------|-------|
| instance   | <a href="#">EnterpriseRouter</a> object | 企业路由器 |
| request_id | String                                  | 请求ID  |

表 4-8 EnterpriseRouter

| 参数                         | 参数类型                                 | 描述                                                       |
|----------------------------|--------------------------------------|----------------------------------------------------------|
| id                         | String                               | 企业路由器实例的ID                                               |
| name                       | String                               | 企业路由器实例名称                                                |
| description                | String                               | 企业路由器实例描述信息                                              |
| state                      | String                               | 运行状态:pending available modifying deleting deleted failed |
| tags                       | Array of <a href="#">Tag</a> objects | 标签信息                                                     |
| charge_mode                | String                               | 计费模式 按需<br>缺省值: <b>postPaid</b>                          |
| created_at                 | String                               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| updated_at                 | String                               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| enterprise_project_id      | String                               | 企业项目ID                                                   |
| project_id                 | String                               | 项目ID                                                     |
| asn                        | Long                                 | 企业路由器实例的BGP AS号                                          |
| enable_default_propagation | Boolean                              | 是否开启默认路由表传播，默认false不开启                                   |

| 参数                                 | 参数类型             | 描述                      |
|------------------------------------|------------------|-------------------------|
| enable_default_association         | Boolean          | 是否开启默认路由表关联，默认false不开启  |
| default_propagation_route_table_id | String           | 默认传播路由表id               |
| default_association_route_table_id | String           | 默认关联路由表id               |
| availability_zone_ids              | Array of strings | 企业路由器所在可用区信息            |
| auto_accept_shared_attachments     | Boolean          | 是否自动接受共享连接创建，默认false不开启 |

表 4-9 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                  |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>_sys_开头属于系统标签，租户不能输入。</li><li>标签key不能为空（长度不能为0）。</li><li>长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                        |

## 请求示例

创建一个企业路由器，名称为“my\_er”，开启默认路由表关联和默认路由表传播功能

POST https://{erouter\_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/instances

```
{
  "instance": {
    "name": "my_er",
    "description": "this is my first enterprise router",
    "asn": 64512,
    "enable_default_association": true,
    "enable_default_propagation": true,
    "tags": [ {
      "key": "key1",
      "value": "value1"
    } ],
    "availability_zone_ids": [ "az1", "az2" ]
  }
}
```

## 响应示例

**状态码：202**

Accepted

```
{
  "instance": {
    "id": "94c2b814-99dc-939a-e811-ae84c61ea3ff",
    "name": "my_er",
    "description": "this is my first enterprise router",
    "asn": 64512,
    "project_id": "08d5a9564a704afda6039ae2babbef3c",
    "enable_default_association": true,
    "enable_default_propagation": true,
    "default_association_route_table_id": "7f7f738f-453c-40b1-be26-28e7b9e390c1",
    "default_propagation_route_table_id": "7f7f738f-453c-40b1-be26-28e7b9e390c1",
    "auto_accept_shared_attachments": false,
    "created_at": "2019-09-06 02:11:13Z",
    "updated_at": "2019-09-06 02:11:13Z",
    "tags": [ {
      "key": "key1",
      "value": "value1"
    } ],
    "enterprise_project_id": 0,
    "availability_zone_ids": [ "az1", "az2" ]
  },
  "request_id": "14c2b814-99dc-939a-e811-ae84c61ea3f4"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个企业路由器，名称为“my\_er”，开启默认路由表关联和默认路由表传播功能

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateEnterpriseRouterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateEnterpriseRouterRequest request = new CreateEnterpriseRouterRequest();
        CreateEnterpriseRouterRequestBody body = new CreateEnterpriseRouterRequestBody();
        List<String> listInstanceAvailabilityZoneIds = new ArrayList<>();
        listInstanceAvailabilityZoneIds.add("az1");
        listInstanceAvailabilityZoneIds.add("az2");
        List<Tag> listInstanceTags = new ArrayList<>();
        listInstanceTags.add(
            new Tag()
                .withKey("key1")
                .withValue("value1")
        );
        CreateEnterpriseRouter instancebody = new CreateEnterpriseRouter();
        instancebody.setName("my_er")
            .withDescription("this is my first enterprise router")
            .withAsn(64512L)
            .withTags(listInstanceTags)
            .withEnableDefaultPropagation(true)
            .withEnableDefaultAssociation(true)
            .withAvailabilityZoneIds(listInstanceAvailabilityZoneIds);
        body.withInstance(instancebody);
        request.withBody(body);
        try {
            CreateEnterpriseRouterResponse response = client.createEnterpriseRouter(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

创建一个企业路由器，名称为“my\_er”，开启默认路由表关联和默认路由表传播功能

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateEnterpriseRouterRequest()
        listAvailabilityZonesInstance = [
            "az1",
            "az2"
        ]
        listTagsInstance = [
            Tag(
                key="key1",
                value="value1"
            )
        ]
        instancebody = CreateEnterpriseRouter(
            name="my_er",
            description="this is my first enterprise router",
            asn=64512,
            tags=listTagsInstance,
            enable_default_propagation=True,
            enable_default_association=True,
            availability_zone_ids=listAvailabilityZonesInstance
        )
        request.body = CreateEnterpriseRouterRequestBody(
            instance=instancebody
        )
        response = client.create_enterprise_router(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

创建一个企业路由器，名称为“my\_er”，开启默认路由表关联和默认路由表传播功能

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
```

```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateEnterpriseRouterRequest{
        var listAvailabilityZoneldsInstance = []string{
            "az1",
            "az2",
        }
        keyTags:= "key1"
        valueTags:= "value1"
        var listTagsInstance = []model.Tag{
            {
                Key: &keyTags,
                Value: &valueTags,
            },
        }
        descriptionInstance:= "this is my first enterprise router"
        enableDefaultPropagationInstance:= true
        enableDefaultAssociationInstance:= true
        instancebody := &model.CreateEnterpriseRouter{
            Name: "my_er",
            Description: &descriptionInstance,
            Asn: int64(64512),
            Tags: &listTagsInstance,
            EnableDefaultPropagation: &enableDefaultPropagationInstance,
            EnableDefaultAssociation: &enableDefaultAssociationInstance,
            AvailabilityZonelds: listAvailabilityZoneldsInstance,
        }
        request.Body = &model.CreateEnterpriseRouterRequestBody{
            Instance: instancebody,
        }
        response, err := client.CreateEnterpriseRouter(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.1.2 更新企业路由器

### 功能介绍

更新企业路由器基本信息。

### 接口约束

除了name和description，其它信息只有在企业路由器实例状态为available的时候才能更新。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/enterprise-router/instances/{er\_id}

表 4-10 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

### 请求参数

表 4-11 请求 Body 参数

| 参数       | 是否必选 | 参数类型                                          | 描述         |
|----------|------|-----------------------------------------------|------------|
| instance | 否    | <a href="#">UpdateEnterpriseRouter</a> object | 更新企业路由器请求体 |

表 4-12 UpdateEnterpriseRouter

| 参数                                 | 是否必选 | 参数类型    | 描述                                                                        |
|------------------------------------|------|---------|---------------------------------------------------------------------------|
| name                               | 否    | String  | 企业路由器实例名称，取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-（中划线）、.（点）<br>最小长度：1<br>最大长度：64 |
| description                        | 否    | String  | 企业路由器实例描述信息<br>最小长度：0<br>最大长度：255                                         |
| enable_default_propagation         | 否    | Boolean | 是否开启默认传播                                                                  |
| enable_default_association         | 否    | Boolean | 是否开启默认关联                                                                  |
| default_propagation_route_table_id | 否    | String  | 默认传播路由表ID                                                                 |
| default_association_route_table_id | 否    | String  | 默认关联路由表ID                                                                 |
| auto_accept_shared_attachments     | 否    | Boolean | 是否自动接受共享连接创建，默认false不开启                                                   |

响应参数

状态码：200

表 4-13 响应 Body 参数

| 参数         | 参数类型                    | 描述    |
|------------|-------------------------|-------|
| instance   | EnterpriseRouter object | 企业路由器 |
| request_id | String                  | 请求ID  |

表 4-14 EnterpriseRouter

| 参数 | 参数类型   | 描述         |
|----|--------|------------|
| id | String | 企业路由器实例的ID |

| 参数                                 | 参数类型                        | 描述                                                       |
|------------------------------------|-----------------------------|----------------------------------------------------------|
| name                               | String                      | 企业路由器实例名称                                                |
| description                        | String                      | 企业路由器实例描述信息                                              |
| state                              | String                      | 运行状态:pending available modifying deleting deleted failed |
| tags                               | Array of <b>Tag</b> objects | 标签信息                                                     |
| charge_mode                        | String                      | 计费模式 按需<br>缺省值: <b>postPaid</b>                          |
| created_at                         | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| updated_at                         | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| enterprise_project_id              | String                      | 企业项目ID                                                   |
| project_id                         | String                      | 项目ID                                                     |
| asn                                | Long                        | 企业路由器实例的BGP AS号                                          |
| enable_default_propagation         | Boolean                     | 是否开启默认路由表传播, 默认false不开启                                  |
| enable_default_association         | Boolean                     | 是否开启默认路由表关联, 默认false不开启                                  |
| default_propagation_route_table_id | String                      | 默认传播路由表id                                                |
| default_association_route_table_id | String                      | 默认关联路由表id                                                |
| availability_zone_ids              | Array of strings            | 企业路由器所在可用区信息                                             |
| auto_accept_shared_attachments     | Boolean                     | 是否自动接受共享连接创建, 默认false不开启                                 |

表 4-15 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

请求示例

更新企业路由器，开启默认路由表关联和默认路由表传播功能，并指定对应的路由表

PUT https://{erouter\_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/instances/94c2b814-99dc-939a-e811-ae84c61ea3ff

```
{
  "instance": {
    "enable_default_association": true,
    "enable_default_propagation": true,
    "default_propagation_route_table_id": "94c2b814-99dc-939a-e811-ae84c61ea3ff",
    "default_association_route_table_id": "94c2b814-99dc-939a-e811-ae84c61ea3ff"
  }
}
```

响应示例

状态码：200

OK

```
{
  "instance": {
```

```
"id" : "94c2b814-99dc-939a-e811-ae84c61ea3ff",
"name" : "my_er",
"description" : "this is my first enterprise router",
"project_id" : "08d5a9564a704afda6039ae2babbe3c",
"state" : "pending",
"asn" : 64512,
"enable_default_association" : true,
"enable_default_propagation" : true,
"default_propagation_route_table_id" : "94c2b814-99dc-939a-e811-ae84c61ea3ff",
"default_association_route_table_id" : "94c2b814-99dc-939a-e811-ae84c61ea3ff",
"auto_accept_shared_attachments" : false,
"availability_zone_ids" : [ "az1" ],
"created_at" : "2019-09-06 02:11:13Z",
"updated_at" : "2019-09-06 02:11:13Z"
},
"request_id" : "14c2b814-99dc-939b-e81c-ae84c61ea3f7"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新企业路由器，开启默认路由表关联和默认路由表传播功能，并指定对应的路由表

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class UpdateEnterpriseRouterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        UpdateEnterpriseRouterRequest request = new UpdateEnterpriseRouterRequest();
        request.withErId("{er_id}");
        UpdateEnterpriseRouterRequestBody body = new UpdateEnterpriseRouterRequestBody();
        UpdateEnterpriseRouter instancebody = new UpdateEnterpriseRouter();
        instancebody.withEnableDefaultPropagation(true)
            .withEnableDefaultAssociation(true)
            .withDefaultPropagationRouteTableId("94c2b814-99dc-939a-e811-ae84c61ea3ff")
            .withDefaultAssociationRouteTableId("94c2b814-99dc-939a-e811-ae84c61ea3ff");
```

```
body.withInstance(instancebody);
request.withBody(body);
try {
    UpdateEnterpriseRouterResponse response = client.updateEnterpriseRouter(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

更新企业路由器，开启默认路由表关联和默认路由表传播功能，并指定对应的路由表

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateEnterpriseRouterRequest()
        request.er_id = "{er_id}"
        instancebody = UpdateEnterpriseRouter(
            enable_default_propagation=True,
            enable_default_association=True,
            default_propagation_route_table_id="94c2b814-99dc-939a-e811-ae84c61ea3ff",
            default_association_route_table_id="94c2b814-99dc-939a-e811-ae84c61ea3ff"
        )
        request.body = UpdateEnterpriseRouterRequestBody(
            instance=instancebody
        )
        response = client.update_enterprise_router(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

更新企业路由器，开启默认路由表关联和默认路由表传播功能，并指定对应的路由表

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.NewErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateEnterpriseRouterRequest{}
    request.ErId = "{er_id}"
    enableDefaultPropagationInstance := true
    enableDefaultAssociationInstance := true
    defaultPropagationRouteTableIdInstance := "94c2b814-99dc-939a-e811-ae84c61ea3ff"
    defaultAssociationRouteTableIdInstance := "94c2b814-99dc-939a-e811-ae84c61ea3ff"
    instancebody := &model.UpdateEnterpriseRouter{
        EnableDefaultPropagation: &enableDefaultPropagationInstance,
        EnableDefaultAssociation: &enableDefaultAssociationInstance,
        DefaultPropagationRouteTableId: &defaultPropagationRouteTableIdInstance,
        DefaultAssociationRouteTableId: &defaultAssociationRouteTableIdInstance,
    }
    request.Body = &model.UpdateEnterpriseRouterRequestBody{
        Instance: instancebody,
    }
    response, err := client.UpdateEnterpriseRouter(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.1.3 查询企业路由器详情

### 功能介绍

查询企业路由器详情

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/instances/{er\_id}

表 4-16 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

### 请求参数

无

### 响应参数

状态码：200

表 4-17 响应 Body 参数

| 参数         | 参数类型                                    | 描述    |
|------------|-----------------------------------------|-------|
| instance   | <a href="#">EnterpriseRouter</a> object | 企业路由器 |
| request_id | String                                  | 请求ID  |

表 4-18 EnterpriseRouter

| 参数                                 | 参数类型                                 | 描述                                                       |
|------------------------------------|--------------------------------------|----------------------------------------------------------|
| id                                 | String                               | 企业路由器实例的ID                                               |
| name                               | String                               | 企业路由器实例名称                                                |
| description                        | String                               | 企业路由器实例描述信息                                              |
| state                              | String                               | 运行状态:pending available modifying deleting deleted failed |
| tags                               | Array of <a href="#">Tag</a> objects | 标签信息                                                     |
| charge_mode                        | String                               | 计费模式 按需<br>缺省值: <b>postPaid</b>                          |
| created_at                         | String                               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| updated_at                         | String                               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| enterprise_project_id              | String                               | 企业项目ID                                                   |
| project_id                         | String                               | 项目ID                                                     |
| asn                                | Long                                 | 企业路由器实例的BGP AS号                                          |
| enable_default_propagation         | Boolean                              | 是否开启默认路由表传播, 默认false不开启                                  |
| enable_default_association         | Boolean                              | 是否开启默认路由表关联, 默认false不开启                                  |
| default_propagation_route_table_id | String                               | 默认传播路由表id                                                |
| default_association_route_table_id | String                               | 默认关联路由表id                                                |
| availability_zone_ids              | Array of strings                     | 企业路由器所在可用区信息                                             |
| auto_accept_shared_attachments     | Boolean                              | 是否自动接受共享连接创建, 默认false不开启                                 |

表 4-19 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                           |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

请求示例

获取指定id的企业路由器详情

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/instances/08d5a9564a704afda6039ae2babbe3c
```

响应示例

状态码：200

OK

```
{
  "instance": {
    "id": "94c2b814-99dc-939a-e811-ae84c61ea3ff",
    "name": "my_er",
    "description": "this is my first enterprise router",
    "project_id": "08d5a9564a704afda6039ae2babbe3c",
    "state": "pending",
    "asn": 64512,
    "enable_default_association": true,
    "enable_default_propagation": true,
  }
}
```

```
"default_association_route_table_id" : "7f7f738f-453c-40b1-be26-28e7b9e390c1",
"default_propagation_route_table_id" : "7f7f738f-453c-40b1-be26-28e7b9e390c1",
"auto_accept_shared_attachments" : false,
"availability_zone_ids" : [ "az1" ],
"created_at" : "2019-09-06 02:11:13Z",
"updated_at" : "2019-09-06 02:11:13Z"
},
"request_id" : "14c2b814-99dc-939b-e81c-ae84c61ea3f7"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ShowEnterpriseRouterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowEnterpriseRouterRequest request = new ShowEnterpriseRouterRequest();
        request.withErId("{er_id}");
        try {
            ShowEnterpriseRouterResponse response = client.showEnterpriseRouter(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowEnterpriseRouterRequest()
        request.er_id = "{er_id}"
        response = client.show_enterprise_router(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).  
Build())  
  
request := &model.ShowEnterpriseRouterRequest{  
    request.ErId = "{er_id}"  
    response, err := client.ShowEnterpriseRouter(request)  
    if err == nil {  
        fmt.Printf("%+v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.1.4 查询企业路由器列表

### 功能介绍

查询企业路由器列表

### 接口约束

- 分页查询使用的参数为marker、limit。marker和limit一起使用时才会生效，单独使用无效。
- 支持单字段排序，排序字段有[id,name,description,created\_at,updated\_at]，不支持多字段排序。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/instances

表 4-20 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

表 4-21 Query 参数

| 参数                    | 是否必选 | 参数类型             | 描述                                                                             |
|-----------------------|------|------------------|--------------------------------------------------------------------------------|
| limit                 | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                  |
| marker                | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b> |
| enterprise_project_id | 否    | Array of strings | 企业项目ID                                                                         |
| state                 | 否    | Array of strings | 资源状态:pending available modifying deleting deleted failed freezed               |
| id                    | 否    | Array of strings | 根据资源ID查询，可同时查询多个。                                                              |
| resource_id           | 否    | Array of strings | 连接对应的资源ID列表                                                                    |
| owned_by_self         | 否    | Boolean          | 过滤资源是否属于当前租户；取值为true时，只查属于当前租户的资源，不包括共享资源；为false时，查询当前租户及共享给该租户的资源；            |
| sort_key              | 否    | Array of strings | 按关键字排序，默认按照id排序，可选值:id name state                                              |
| sort_dir              | 否    | Array of strings | 返回结果按照升序或降序排列，默认为asc,降序为desc                                                   |

请求参数

无

响应参数

状态码：200

表 4-22 响应 Body 参数

| 参数         | 参数类型                                     | 描述      |
|------------|------------------------------------------|---------|
| instances  | Array of <b>EnterpriseRouter</b> objects | 企业路由器列表 |
| page_info  | <b>PageInfo</b> object                   | 分页查询的信息 |
| request_id | String                                   | 请求ID    |

表 4-23 EnterpriseRouter

| 参数                                 | 参数类型                        | 描述                                                       |
|------------------------------------|-----------------------------|----------------------------------------------------------|
| id                                 | String                      | 企业路由器实例的ID                                               |
| name                               | String                      | 企业路由器实例名称                                                |
| description                        | String                      | 企业路由器实例描述信息                                              |
| state                              | String                      | 运行状态:pending available modifying deleting deleted failed |
| tags                               | Array of <b>Tag</b> objects | 标签信息                                                     |
| charge_mode                        | String                      | 计费模式 按需<br>缺省值： <b>postPaid</b>                          |
| created_at                         | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| updated_at                         | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| enterprise_project_id              | String                      | 企业项目ID                                                   |
| project_id                         | String                      | 项目ID                                                     |
| asn                                | Long                        | 企业路由器实例的BGP AS号                                          |
| enable_default_propagation         | Boolean                     | 是否开启默认路由表传播，默认false不开启                                   |
| enable_default_association         | Boolean                     | 是否开启默认路由表关联，默认false不开启                                   |
| default_propagation_route_table_id | String                      | 默认传播路由表id                                                |

| 参数                                 | 参数类型             | 描述                      |
|------------------------------------|------------------|-------------------------|
| default_association_route_table_id | String           | 默认关联路由表id               |
| availability_zone_ids              | Array of strings | 企业路由器所在可用区信息            |
| auto_accept_shared_attachments     | Boolean          | 是否自动接受共享连接创建，默认false不开启 |

表 4-24 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

表 4-25 PageInfo

| 参数          | 参数类型   | 描述                    |
|-------------|--------|-----------------------|
| next_marker | String | 下一个marker标识，为空时表示最后一页 |

| 参数            | 参数类型    | 描述        |
|---------------|---------|-----------|
| current_count | Integer | 当前列表中资源数量 |

## 请求示例

- 分页查询2两条记录

GET https://{erouter\_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/instances?limit=2&marker=1

- 全量查询

GET https://{erouter\_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/instances

## 响应示例

状态码：200

OK

```
{
  "instances": [ {
    "id": "94c2b814-99dc-939a-e811-ae84c61ea3ff",
    "name": "my_er",
    "description": "this is my first enterprise router",
    "asn": 64512,
    "project_id": "08d5a9564a704afda6039ae2babbef3c",
    "enable_default_association": true,
    "enable_default_propagation": true,
    "default_association_route_table_id": "7f7f738f-453c-40b1-be26-28e7b9e390c1",
    "default_propagation_route_table_id": "7f7f738f-453c-40b1-be26-28e7b9e390c1",
    "auto_accept_shared_attachments": false,
    "created_at": "2019-09-06 02:11:13Z",
    "updated_at": "2019-09-06 02:11:13Z",
    "tags": [ {
      "key": "key",
      "value": "value"
    } ],
    "enterprise_project_id": 0,
    "availability_zone_ids": [ "az1", "az2" ]
  } ],
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9",
  "page_info": {
    "next_marker": "2",
    "current_count": 1
  }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
```

```
import com.huaweicloud.sdk.er.v3.model.*;

public class ListEnterpriseRoutersSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEnterpriseRoutersRequest request = new ListEnterpriseRoutersRequest();
        try {
            ListEnterpriseRoutersResponse response = client.listEnterpriseRouters(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = ListEnterpriseRoutersRequest()
    response = client.list_enterprise_routers(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEnterpriseRoutersRequest{}
    response, err := client.ListEnterpriseRouters(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.1.5 更新企业路由器的可用区信息

功能介绍

更新企业路由器的可用区信息，企业路由器实例状态为available的时候才能更新。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project\_id}/enterprise-router/instances/{er\_id}/change-availability-zone-ids

表 4-26 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

请求参数

表 4-27 请求 Body 参数

| 参数                    | 是否必选 | 参数类型             | 描述                                  |
|-----------------------|------|------------------|-------------------------------------|
| availability_zone_ids | 是    | Array of strings | 企业路由器实例所在的可用区<br>最小长度：1<br>最大长度：256 |

响应参数

状态码：202

表 4-28 响应 Body 参数

| 参数         | 参数类型                                    | 描述    |
|------------|-----------------------------------------|-------|
| instance   | <a href="#">EnterpriseRouter</a> object | 企业路由器 |
| request_id | String                                  | 请求ID  |

表 4-29 EnterpriseRouter

| 参数                                 | 参数类型                        | 描述                                                       |
|------------------------------------|-----------------------------|----------------------------------------------------------|
| id                                 | String                      | 企业路由器实例的ID                                               |
| name                               | String                      | 企业路由器实例名称                                                |
| description                        | String                      | 企业路由器实例描述信息                                              |
| state                              | String                      | 运行状态:pending available modifying deleting deleted failed |
| tags                               | Array of <b>Tag</b> objects | 标签信息                                                     |
| charge_mode                        | String                      | 计费模式 按需<br>缺省值: <b>postPaid</b>                          |
| created_at                         | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| updated_at                         | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| enterprise_project_id              | String                      | 企业项目ID                                                   |
| project_id                         | String                      | 项目ID                                                     |
| asn                                | Long                        | 企业路由器实例的BGP AS号                                          |
| enable_default_propagation         | Boolean                     | 是否开启默认路由表传播，默认false不开启                                   |
| enable_default_association         | Boolean                     | 是否开启默认路由表关联，默认false不开启                                   |
| default_propagation_route_table_id | String                      | 默认传播路由表id                                                |
| default_association_route_table_id | String                      | 默认关联路由表id                                                |
| availability_zone_ids              | Array of strings            | 企业路由器所在可用区信息                                             |
| auto_accept_shared_attachments     | Boolean                     | 是否自动接受共享连接创建，默认false不开启                                  |

表 4-30 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                           |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

请求示例

更新企业路由器的可用区

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/instances/cccf35ad-fd0d-4043-aab2-dc28ae56f967/change-availability-zone-ids
{
  "availability_zone_ids": [ "az1" ]
}
```

响应示例

状态码：202

Accepted

```
{
  "instance": {
    "id": "94c2b814-99dc-939a-e811-ae84c61ea3ff",
    "name": "my_er",
    "description": "this is my first enterprise router",
    "project_id": "08d5a9564a704afda6039ae2babbef3c",
```

```
"state": "pending",
"asn": 64512,
"enable_default_association": true,
"enable_default_propagation": true,
"default_association_route_table_id": "7f7f738f-453c-40b1-be26-28e7b9e390c1",
"default_propagation_route_table_id": "7f7f738f-453c-40b1-be26-28e7b9e390c1",
"availability_zone_ids": [ "az1" ],
"created_at": "2019-09-06 02:11:13Z",
"updated_at": "2019-09-06 02:11:13Z"
},
"request_id": "5f7f738f-453c-40b1-be26-28e7b9e390c2"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新企业路由器的可用区

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class ChangeAvailabilityZoneSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();

        ChangeAvailabilityZoneRequest request = new ChangeAvailabilityZoneRequest();
        request.withErId("{er_id}");
        EnterpriseRouterAZ body = new EnterpriseRouterAZ();
        List<String> listbodyAvailabilityZonelds = new ArrayList<>();
        listbodyAvailabilityZonelds.add("az1");
        body.withAvailabilityZonelds(listbodyAvailabilityZonelds);
        request.withBody(body);
        try {
            ChangeAvailabilityZoneResponse response = client.changeAvailabilityZone(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

### 更新企业路由器的可用区

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ChangeAvailabilityZoneRequest()
        request.er_id = "{er_id}"
        listAvailabilityZoneIdsbody = [
            "az1"
        ]
        request.body = EnterpriseRouterAZ(
            availability_zone_ids=listAvailabilityZoneIdsbody
        )
        response = client.change_availability_zone(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新企业路由器的可用区

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ChangeAvailabilityZoneRequest{}
    request.ErId = "{er_id}"
    var listAvailabilityZonesbody = []string{
        "az1",
    }
    request.Body = &model.EnterpriseRouterAz{
        AvailabilityZones: listAvailabilityZonesbody,
    }
    response, err := client.ChangeAvailabilityZone(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.1.6 删除企业路由器

### 功能介绍

删除企业路由器。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{project\_id}/enterprise-router/instances/{er\_id}

表 4-31 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

### 请求参数

无

### 响应参数

无

### 请求示例

删除企业路由器

```
DELETE https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/instances/94c2b814-99dc-939a-e811-ae84c61ea3ff
```

### 响应示例

无

### SDK 代码示例

SDK代码示例如下。

#### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DeleteEnterpriseRouterSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteEnterpriseRouterRequest request = new DeleteEnterpriseRouterRequest();
        request.withErId("{er_id}");
        try {
            DeleteEnterpriseRouterResponse response = client.deleteEnterpriseRouter(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(ErRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DeleteEnterpriseRouterRequest()
    request.er_id = "{er_id}"
    response = client.delete_enterprise_router(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.NewErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteEnterpriseRouterRequest{}
    request.ErId = "{er_id}"
    response, err := client.DeleteEnterpriseRouter(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

# 4.2 VPC 连接

## 4.2.1 创建 VPC 连接

### 功能介绍

给ER实例创建VPC连接。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/vpc-attachments

表 4-32 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

### 请求参数

表 4-33 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-34 请求 Body 参数

| 参数             | 是否必选 | 参数类型                              | 描述      |
|----------------|------|-----------------------------------|---------|
| vpc_attachment | 是    | VpcAttachmentCreateRequest object | VPC类型连接 |

表 4-35 VpcAttachmentCreateRequest

| 参数                     | 是否必选 | 参数类型                 | 描述                                                                      |
|------------------------|------|----------------------|-------------------------------------------------------------------------|
| vpc_id                 | 是    | String               | VPC的id，取值范围：最大长度36字节，带“-”连字符的UUID格式                                     |
| name                   | 是    | String               | VPC连接名字，取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)<br>最小长度：1<br>最大长度：64 |
| virsubnet_id           | 是    | String               | VPC子网id，取值范围：最大长度36字节，带“-”连字符的UUID格式                                    |
| description            | 否    | String               | 描述信息，取值范围：0-255个字符，不能包含“<”和“>”<br>最小长度：0<br>最大长度：255                    |
| auto_create_vpc_routes | 否    | Boolean              | 默认为false，为true表示自动为vpc配置指向企业路由器的路由<br>缺省值：false                         |
| tags                   | 否    | Array of Tag objects | 标签信息                                                                    |

表 4-36 Tag

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                           |
|-------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | 否    | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | 否    | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

响应参数

状态码：202

表 4-37 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-38 响应 Body 参数

| 参数             | 参数类型                               | 描述      |
|----------------|------------------------------------|---------|
| vpc_attachment | <b>VpcAttachmentDetails</b> object | VPC类型连接 |
| request_id     | String                             | 请求id    |

表 4-39 VpcAttachmentDetails

| 参数                     | 参数类型                        | 描述                                                                                                                         |
|------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------|
| id                     | String                      | VPC连接ID                                                                                                                    |
| name                   | String                      | VPC连接名称                                                                                                                    |
| vpc_id                 | String                      | VPC id                                                                                                                     |
| virsubnet_id           | String                      | VPC子网id                                                                                                                    |
| auto_create_vpc_routes | Boolean                     | 默认为false,当设置true时，会自动为VPC配置10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16三条路由，下一跳指向企业路由器。<br>缺省值： <b>false</b>                  |
| state                  | String                      | VPC连接状态:pending available modifying deleting deleted failed initiating_request rejected pending_acceptance freezed invalid |
| created_at             | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                           |
| updated_at             | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                           |
| tags                   | Array of <b>Tag</b> objects | 标签信息                                                                                                                       |
| description            | String                      | VPC连接描述信息                                                                                                                  |
| project_id             | String                      | 项目ID                                                                                                                       |
| vpc_project_id         | String                      | vpc所属项目ID                                                                                                                  |

表 4-40 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

请求示例

创建一个VPC连接，名称为“vpc-atta”，并指定连接的VPC和子网资源

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/4ab54142-7c92-48ad-8288-77727a231052/vpc-attachments
{
  "vpc_attachment": {
    "name": "vpc-atta",
    "vpc_id": "b715e131-3371-4e17-a2de-4f669e24439a",
    "virsubnet_id": "aacdc21d-90f9-45ef-ab48-80ec1bbe15b8"
  }
}
```

响应示例

状态码：202

Accepted

```
{
  "vpc_attachment": {
```

```
"id" : "6f83b848-8331-4271-ac0c-ef94b7686402",
"name" : "vpc-atta",
"vpc_id" : "b715e131-3371-4e17-a2de-4f669e24439a",
"virsubnet_id" : "aacdc21d-90f9-45ef-ab48-80ec1bbe15b8",
"project_id" : "08d5a9564a704afda6039ae2babbe3c",
"state" : "pending",
"auto_create_vpc_routes" : false,
"created_at" : "2020-03-11T15:13:31Z",
"updated_at" : "2020-03-11T15:13:31Z"
},
"request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建一个VPC连接，名称为“vpc-atta”，并指定连接的VPC和子网资源

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class CreateVpcAttachmentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateVpcAttachmentRequest request = new CreateVpcAttachmentRequest();
        request.withErId("{er_id}");
        CreateVpcAttachmentBody body = new CreateVpcAttachmentBody();
        VpcAttachmentCreateRequest vpcAttachmentbody = new VpcAttachmentCreateRequest();
        vpcAttachmentbody.withVpcId("b715e131-3371-4e17-a2de-4f669e24439a")
            .withName("vpc-atta")
            .withVirsubnetId("aacdc21d-90f9-45ef-ab48-80ec1bbe15b8");
        body.withVpcAttachment(vpcAttachmentbody);
        request.withBody(body);
        try {
            CreateVpcAttachmentResponse response = client.createVpcAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
```

```
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

## Python

创建一个VPC连接，名称为“vpc-atta”，并指定连接的VPC和子网资源

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateVpcAttachmentRequest()
        request.er_id = "{er_id}"
        vpcAttachmentbody = VpcAttachmentCreateRequest(
            vpc_id="b715e131-3371-4e17-a2de-4f669e24439a",
            name="vpc-atta",
            virsubnet_id="aacdc21d-90f9-45ef-ab48-80ec1bbe15b8"
        )
        request.body = CreateVpcAttachmentBody(
            vpc_attachment=vpcAttachmentbody
        )
        response = client.create_vpc_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

创建一个VPC连接，名称为“vpc-atta”，并指定连接的VPC和子网资源

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateVpcAttachmentRequest{}
    request.ErId = "{er_id}"
    vpcAttachmentbody := &model.VpcAttachmentCreateRequest{
        VpcId: "b715e131-3371-4e17-a2de-4f669e24439a",
        Name: "vpc-atta",
        VirsubnetId: "aacdc21d-90f9-45ef-ab48-80ec1bbe15b8",
    }
    request.Body = &model.CreateVpcAttachmentBody{
        VpcAttachment: vpcAttachmentbody,
    }
    response, err := client.CreateVpcAttachment(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.2.2 更新 VPC 连接基本信息

### 功能介绍

修改VPC连接基本信息。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/enterprise-router/{er\_id}/vpc-attachments/{vpc\_attachment\_id}

表 4-41 路径参数

| 参数                | 是否必选 | 参数类型   | 描述        |
|-------------------|------|--------|-----------|
| project_id        | 是    | String | 项目ID      |
| er_id             | 是    | String | 企业路由器实例ID |
| vpc_attachment_id | 是    | String | VPC连接ID   |

### 请求参数

表 4-42 请求 Body 参数

| 参数             | 是否必选 | 参数类型                                           | 描述           |
|----------------|------|------------------------------------------------|--------------|
| vpc_attachment | 否    | <a href="#">UpdateVpcAttachmentBody</a> object | 更新vpc基本信息请求体 |

表 4-43 UpdateVpcAttachmentBody

| 参数          | 是否必选 | 参数类型   | 描述                                                                               |
|-------------|------|--------|----------------------------------------------------------------------------------|
| description | 否    | String | VPC连接描述信息，取值范围：<br>最大长度36字节，带“-”连字符的UUID格式<br>最小长度： <b>0</b><br>最大长度： <b>255</b> |

| 参数   | 是否必选 | 参数类型   | 描述                                                                      |
|------|------|--------|-------------------------------------------------------------------------|
| name | 否    | String | VPC连接名称，取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)<br>最小长度：1<br>最大长度：64 |

响应参数

状态码：200

表 4-44 响应 Body 参数

| 参数             | 参数类型                                        | 描述      |
|----------------|---------------------------------------------|---------|
| vpc_attachment | <a href="#">VpcAttachmentDetails</a> object | VPC类型连接 |
| request_id     | String                                      | 请求id    |

表 4-45 VpcAttachmentDetails

| 参数                     | 参数类型    | 描述                                                                                                                        |
|------------------------|---------|---------------------------------------------------------------------------------------------------------------------------|
| id                     | String  | VPC连接ID                                                                                                                   |
| name                   | String  | VPC连接名称                                                                                                                   |
| vpc_id                 | String  | VPC id                                                                                                                    |
| virsubnet_id           | String  | VPC子网id                                                                                                                   |
| auto_create_vpc_routes | Boolean | 默认为false,当设置true时，会自动为VPC配置10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16三条路由，下一跳指向企业路由器。<br>缺省值：false                         |
| state                  | String  | VPC连接状态:pending available modifying deleting deleted failed initiating_request rejected pending_acceptance frezed invalid |
| created_at             | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                          |
| updated_at             | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                          |

| 参数             | 参数类型                        | 描述        |
|----------------|-----------------------------|-----------|
| tags           | Array of <b>Tag</b> objects | 标签信息      |
| description    | String                      | VPC连接描述信息 |
| project_id     | String                      | 项目ID      |
| vpc_project_id | String                      | vpc所属项目ID |

表 4-46 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

请求示例

更新VPC连接的信息，修改连接名称为 “new-vpc-attach”

```
PUT https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/f1a28dfd-186f-4625-b6b1-f05e5e8609c0/vpc-attachments/3b9724e9-49ec-4d21-9191-3d703133b910
{
  "vpc_attachment": {
```

```
    "name" : "new-vpc-attach"  
  }  
}
```

## 响应示例

状态码：200

OK

```
{  
  "vpc_attachment" : {  
    "id" : "3b9724e9-49ec-4d21-9191-3d703133b910",  
    "name" : "new-vpc-attach",  
    "vpc_id" : "b715e131-3371-4e17-a2de-4f669e24439a",  
    "vsubnet_id" : "aacdc21d-90f9-45ef-ab48-80ec1bbe15b8",  
    "auto_create_vpc_routes" : false,  
    "project_id" : "08d5a9564a704afda6039ae2babbe3c",  
    "state" : "available",  
    "created_at" : "2020-03-11T15:13:31Z",  
    "updated_at" : "2020-03-11T15:13:31Z"  
  },  
  "request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新VPC连接的信息，修改连接名称为“new-vpc-attach”

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.er.v3.region.ErRegion;  
import com.huaweicloud.sdk.er.v3.*;  
import com.huaweicloud.sdk.er.v3.model.*;  
  
public class UpdateVpcAttachmentSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        ErClient client = ErClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))  
            .build();
```

```
UpdateVpcAttachmentRequest request = new UpdateVpcAttachmentRequest();
request.withErId("{er_id}");
request.withVpcAttachmentId("{vpc_attachment_id}");
UpdateVpcAttachmentRequestBody body = new UpdateVpcAttachmentRequestBody();
UpdateVpcAttachmentBody vpcAttachmentbody = new UpdateVpcAttachmentBody();
vpcAttachmentbody.withName("new-vpc-attach");
body.withVpcAttachment(vpcAttachmentbody);
request.withBody(body);
try {
    UpdateVpcAttachmentResponse response = client.updateVpcAttachment(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

更新VPC连接的信息，修改连接名称为“new-vpc-attach”

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateVpcAttachmentRequest()
        request.er_id = "{er_id}"
        request.vpc_attachment_id = "{vpc_attachment_id}"
        vpcAttachmentbody = UpdateVpcAttachmentBody(
            name="new-vpc-attach"
        )
        request.body = UpdateVpcAttachmentRequestBody(
            vpc_attachment=vpcAttachmentbody
        )
        response = client.update_vpc_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

更新VPC连接的信息，修改连接名称为“new-vpc-attach”

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateVpcAttachmentRequest{}
    request.ErId = "{er_id}"
    request.VpcAttachmentId = "{vpc_attachment_id}"
    nameVpcAttachment := "new-vpc-attach"
    vpcAttachmentbody := &model.UpdateVpcAttachmentBody{
        Name: &nameVpcAttachment,
    }
    request.Body = &model.UpdateVpcAttachmentRequestBody{
        VpcAttachment: vpcAttachmentbody,
    }
    response, err := client.UpdateVpcAttachment(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

### 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

### 错误码

请参见[错误码](#)。

## 4.2.3 查询 VPC 连接详情

### 功能介绍

查询VPC连接详情

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/vpc-attachments/  
{vpc\_attachment\_id}

表 4-47 路径参数

| 参数                | 是否必选 | 参数类型   | 描述        |
|-------------------|------|--------|-----------|
| project_id        | 是    | String | 项目ID      |
| er_id             | 是    | String | 企业路由器实例ID |
| vpc_attachment_id | 是    | String | VPC连接ID   |

### 请求参数

无

### 响应参数

状态码：200

表 4-48 响应 Body 参数

| 参数             | 参数类型                               | 描述      |
|----------------|------------------------------------|---------|
| vpc_attachment | <b>VpcAttachmentDetails</b> object | VPC类型连接 |
| request_id     | String                             | 请求id    |

表 4-49 VpcAttachmentDetails

| 参数                     | 参数类型                        | 描述                                                                                                                         |
|------------------------|-----------------------------|----------------------------------------------------------------------------------------------------------------------------|
| id                     | String                      | VPC连接ID                                                                                                                    |
| name                   | String                      | VPC连接名称                                                                                                                    |
| vpc_id                 | String                      | VPC id                                                                                                                     |
| virsubnet_id           | String                      | VPC子网id                                                                                                                    |
| auto_create_vpc_routes | Boolean                     | 默认为false,当设置true时，会自动为VPC配置10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16三条路由，下一跳指向企业路由器。<br>缺省值： <b>false</b>                  |
| state                  | String                      | VPC连接状态:pending available modifying deleting deleted failed initiating_request rejected pending_acceptance freezed invalid |
| created_at             | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                           |
| updated_at             | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                           |
| tags                   | Array of <b>Tag</b> objects | 标签信息                                                                                                                       |
| description            | String                      | VPC连接描述信息                                                                                                                  |
| project_id             | String                      | 项目ID                                                                                                                       |
| vpc_project_id         | String                      | vpc所属项目ID                                                                                                                  |

表 4-50 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                   |
|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>_sys_ 开头属于系统标签，租户不能输入。</li><li>标签key不能为空（长度不能为0）。</li><li>长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                         |

## 请求示例

### 查询VPC连接详情

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/212d3f0b-8496-472d-bc99-05a7c96d6655/vpc-attachments/b70aee08-c671-4cad-9fd5-7381d163bcc8
```

## 响应示例

状态码：200

OK

```
{
  "vpc_attachment": {
    "id": "b70aee08-c671-4cad-9fd5-7381d163bcc8",
    "name": "vpc-attach",
    "vpc_id": "b715e131-3371-4e17-a2de-4f669e24439a",
    "virsubnet_id": "aacdc21d-90f9-45ef-ab48-80ec1bbe15b8",
    "project_id": "08d5a9564a704afda6039ae2babbe3c",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  }
}
```

```
},  
"request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.er.v3.region.ErRegion;  
import com.huaweicloud.sdk.er.v3.*;  
import com.huaweicloud.sdk.er.v3.model.*;  
  
public class ShowVpcAttachmentSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        ErClient client = ErClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ShowVpcAttachmentRequest request = new ShowVpcAttachmentRequest();  
        request.withErId("{er_id}");  
        request.withVpcAttachmentId("{vpc_attachment_id}");  
        try {  
            ShowVpcAttachmentResponse response = client.showVpcAttachment(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowVpcAttachmentRequest()
        request.er_id = "{er_id}"
        request.vpc_attachment_id = "{vpc_attachment_id}"
        response = client.show_vpc_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ShowVpcAttachmentRequest{}
request.ErId = "{er_id}"
request.VpcAttachmentId = "{vpc_attachment_id}"
response, err := client.ShowVpcAttachment(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.2.4 查询 VPC 连接列表

### 功能介绍

查询企业路由器实例下的VPC连接列表。

### 接口约束

- 支持过滤查询，过滤条件有id，state，enterprise\_project\_id，vpc\_id，过滤条件可以重复和组合。
- 支持分页查询，limit和marker组合实现分页查询。
- 支持单字段排序功能，排序字段有[id,name,description,created\_at,updated\_at]，不支持多字段排序。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/vpc-attachments

表 4-51 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

表 4-52 Query 参数

| 参数       | 是否必选 | 参数类型             | 描述                                                                                                              |
|----------|------|------------------|-----------------------------------------------------------------------------------------------------------------|
| limit    | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                                                   |
| marker   | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b>                                  |
| state    | 否    | Array of strings | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request invalid |
| id       | 否    | Array of strings | 根据资源ID查询，可同时查询多个。                                                                                               |
| sort_key | 否    | Array of strings | 按关键字排序，默认按照id排序，可选值:id name state                                                                               |
| sort_dir | 否    | Array of strings | 返回结果按照升序或降序排列，默认为asc,降序为desc                                                                                    |
| vpc_id   | 否    | Array of strings | VPC id                                                                                                          |

请求参数

无

响应参数

状态码：200

表 4-53 响应 Body 参数

| 参数              | 参数类型                                                  | 描述      |
|-----------------|-------------------------------------------------------|---------|
| vpc_attachments | Array of <a href="#">VpcAttachmentDetails</a> objects | VPC连接列表 |
| page_info       | <a href="#">PageInfo</a> object                       | 分页查询的信息 |
| request_id      | String                                                | 请求ID    |

表 4-54 VpcAttachmentDetails

| 参数                     | 参数类型                                 | 描述                                                                                                                         |
|------------------------|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| id                     | String                               | VPC连接ID                                                                                                                    |
| name                   | String                               | VPC连接名称                                                                                                                    |
| vpc_id                 | String                               | VPC id                                                                                                                     |
| virsubnet_id           | String                               | VPC子网id                                                                                                                    |
| auto_create_vpc_routes | Boolean                              | 默认为false,当设置true时，会自动为VPC配置10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16三条路由，下一跳指向企业路由器。<br>缺省值： <b>false</b>                  |
| state                  | String                               | VPC连接状态:pending available modifying deleting deleted failed initiating_request rejected pending_acceptance freezed invalid |
| created_at             | String                               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                           |
| updated_at             | String                               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                           |
| tags                   | Array of <a href="#">Tag</a> objects | 标签信息                                                                                                                       |
| description            | String                               | VPC连接描述信息                                                                                                                  |
| project_id             | String                               | 项目ID                                                                                                                       |
| vpc_project_id         | String                               | vpc所属项目ID                                                                                                                  |

表 4-55 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                  |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>_sys_开头属于系统标签，租户不能输入。</li><li>标签key不能为空（长度不能为0）。</li><li>长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                        |

表 4-56 PageInfo

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| next_marker   | String  | 下一个marker标识，为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量             |

## 请求示例

- 分页查询VPC连接信息

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/vpc-attachments?limit=10
```

- 查询全量VPC连接信息

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/vpc-attachments
```

## 响应示例

状态码：200

OK

```
{
  "vpc_attachments": [ {
    "id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "name": "vpc-attach-01",
    "vpc_id": "b715e131-3371-4e17-a2de-4f669e24439a",
    "virsubnet_id": "aacdc21d-90f9-45ef-ab48-80ec1bbe15b8",
    "project_id": "08d5a9564a704afda6039ae2babbef3c",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  }, {
    "id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "name": "vpc-attach-01",
    "vpc_id": "4b8567f6-358f-4a7f-8cd3-3cbb82c0b25f",
    "virsubnet_id": "2b55f334-a15e-43a9-ab11-b34c2dbb6fac",
    "project_id": "08d5a9564a704afda6039ae2babbef3c",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  } ],
  "page_info": {
    "next_marker": "1",
    "current_count": 2
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListVpcAttachmentsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
ListVpcAttachmentsRequest request = new ListVpcAttachmentsRequest();
request.setErId("{er_id}");
try {
    ListVpcAttachmentsResponse response = client.listVpcAttachments(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListVpcAttachmentsRequest()
        request.set_er_id("{er_id}")
        response = client.list_vpc_attachments(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListVpcAttachmentsRequest{}
    request.ErId = "{er_id}"
    response, err := client.ListVpcAttachments(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.2.5 删除 VPC 连接

### 功能介绍

删除VPC连接。

## 接口约束

VPC连接状态为available，deleting和failed的时候才能删除。

## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/enterprise-router/{er\_id}/vpc-attachments/  
{vpc\_attachment\_id}

表 4-57 路径参数

| 参数                | 是否必选 | 参数类型   | 描述        |
|-------------------|------|--------|-----------|
| project_id        | 是    | String | 项目ID      |
| er_id             | 是    | String | 企业路由器实例ID |
| vpc_attachment_id | 是    | String | VPC连接ID   |

## 请求参数

无

## 响应参数

无

## 请求示例

删除VPC连接

```
DELETE https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/  
4ba931b6-5273-4ed9-8eeb-484d16a4786f/vpc-attachments/b70aee08-c671-4cad-9fd5-7381d163bcc8
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DeleteVpcAttachmentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteVpcAttachmentRequest request = new DeleteVpcAttachmentRequest();
        request.withErId("{er_id}");
        request.withVpcAttachmentId("{vpc_attachment_id}");
        try {
            DeleteVpcAttachmentResponse response = client.deleteVpcAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = ErClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(ErRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteVpcAttachmentRequest()
    request.er_id = "{er_id}"
    request.vpc_attachment_id = "{vpc_attachment_id}"
    response = client.delete_vpc_attachment(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteVpcAttachmentRequest{
        request.ErId = "{er_id}"
        request.VpcAttachmentId = "{vpc_attachment_id}"
    }
    response, err := client.DeleteVpcAttachment(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

# 4.3 其他连接

## 4.3.1 更新连接基本信息

### 功能介绍

修改连接基本信息。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/enterprise-router/{er\_id}/attachments/{attachment\_id}

表 4-58 路径参数

| 参数            | 是否必选 | 参数类型   | 描述        |
|---------------|------|--------|-----------|
| project_id    | 是    | String | 项目ID      |
| er_id         | 是    | String | 企业路由器实例ID |
| attachment_id | 是    | String | 连接ID      |

### 请求参数

表 4-59 请求 Body 参数

| 参数         | 是否必选 | 参数类型                                        | 描述       |
|------------|------|---------------------------------------------|----------|
| attachment | 否    | <a href="#">UpdateAttachmentBody</a> object | 更新连接基本信息 |

表 4-60 UpdateAttachmentBody

| 参数          | 是否必选 | 参数类型   | 描述                                                                   |
|-------------|------|--------|----------------------------------------------------------------------|
| description | 否    | String | 连接描述信息，取值范围：最大长度36字节，带“-”连字符的UUID格式<br>最小长度：0<br>最大长度：255            |
| name        | 否    | String | 连接名称，取值范围：1-64个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)<br>最小长度：1<br>最大长度：64 |

响应参数

状态码：200

表 4-61 响应 Body 参数

| 参数         | 参数类型                      | 描述       |
|------------|---------------------------|----------|
| attachment | AttachmentResponse object | 连接基本信息详情 |
| request_id | String                    | 请求id     |

表 4-62 AttachmentResponse

| 参数          | 参数类型   | 描述                                                                                                                      |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------|
| name        | String | 连接名字                                                                                                                    |
| id          | String | 连接ID                                                                                                                    |
| description | String | 描述信息                                                                                                                    |
| state       | String | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request freezed invalid |
| created_at  | String | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                        |
| updated_at  | String | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                        |

| 参数                  | 参数类型                        | 描述                                                                                                                                                                                                                                    |
|---------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tags                | Array of <b>Tag</b> objects | 企业路由器关联tag                                                                                                                                                                                                                            |
| project_id          | String                      | 项目ID                                                                                                                                                                                                                                  |
| resource_id         | String                      | 内部连接关联的资源ID                                                                                                                                                                                                                           |
| resource_type       | String                      | 内部连接关联的资源类型: <ul style="list-style-type: none"><li>• vgw: 云专线的虚拟网关</li><li>• vpn: vpn网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering ) ” 连接</li><li>• ecn: 企业连接网</li><li>• vpc: 虚拟私有云</li><li>• cfw: 云防火墙</li></ul> |
| resource_project_id | String                      | 资源所属项目ID                                                                                                                                                                                                                              |

表 4-63 Tag

| 参数  | 参数类型   | 描述                                                                                                                                                                                                                                                                        |
|-----|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key | String | 标签键, 规则: <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格, 以及字符: 下划线“_”, 点“.”, 冒号“:”, 等号“=”, 加号“+”, 中划线“-”, “@”。</li><li>• _sys_开头属于系统标签, 租户不能输入。</li><li>• 标签key不能为空 ( 长度不能为0 ) 。</li><li>• 长度在1-128之间。</li></ul> 最小长度: <b>1</b><br>最大长度: <b>128</b> |

| 参数    | 参数类型   | 描述                                                                                                                                                                                                           |
|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255 |

## 请求示例

更新企业路由器内其他类型连接（非VPC连接）的名称

```
PUT https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/  
212d3f0b-8496-472d-bc99-05a7c96d6655/attachments/b70aee08-c671-4cad-9fd5-7381d163bcc8
```

```
{  
  "attachment": {  
    "name": "attachment",  
    "description": "attachment"  
  }  
}
```

## 响应示例

状态码：200

OK

```
{  
  "attachment": {  
    "id": "b70aee08-c671-4cad-9fd5-7381d163bcc8",  
    "name": "attachment",  
    "description": "attachment",  
    "state": "available",  
    "created_at": "2020-03-11T15:13:31Z",  
    "updated_at": "2020-03-11T15:13:31Z",  
    "project_id": "08d5a9564a704afda6039ae2babbef3c",  
    "resource_id": "b715e131-3371-4e17-a2de-4f669e24439a",  
    "resource_type": "vpc",  
    "resource_project_id": "0605767a9980d5762fbcc00b3537e757"  
  },  
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新企业路由器内其他类型连接（非VPC连接）的名称

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class UpdateAttachmentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateAttachmentRequest request = new UpdateAttachmentRequest();
        request.withErId("{er_id}");
        request.withAttachmentId("{attachment_id}");
        UpdateAttachmentRequestBody body = new UpdateAttachmentRequestBody();
        UpdateAttachmentBody attachmentbody = new UpdateAttachmentBody();
        attachmentbody.withDescription("attachment")
            .withName("attachment");
        body.withAttachment(attachmentbody);
        request.withBody(body);
        try {
            UpdateAttachmentResponse response = client.updateAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

更新企业路由器内其他类型连接（非VPC连接）的名称

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateAttachmentRequest()
        request.er_id = "{er_id}"
        request.attachment_id = "{attachment_id}"
        attachmentbody = UpdateAttachmentBody(
            description="attachment",
            name="attachment"
        )
        request.body = UpdateAttachmentRequestBody(
            attachment=attachmentbody
        )
        response = client.update_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 更新企业路由器内其他类型连接（非VPC连接）的名称

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
```

```
er.ErClientBuilder().
    WithRegion(region.ValueOf("<YOUR REGION>")).
    WithCredential(auth).
    Build())

request := &model.UpdateAttachmentRequest{}
request.ErId = "{er_id}"
request.AttachmentId = "{attachment_id}"
descriptionAttachment:= "attachment"
nameAttachment:= "attachment"
attachmentbody := &model.UpdateAttachmentBody{
    Description: &descriptionAttachment,
    Name: &nameAttachment,
}
request.Body = &model.UpdateAttachmentRequestBody{
    Attachment: attachmentbody,
}
response, err := client.UpdateAttachment(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

### 4.3.2 查询连接详情

## 功能介绍

查询连接详情

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/attachments/{attachment\_id}

表 4-64 路径参数

| 参数            | 是否必选 | 参数类型   | 描述        |
|---------------|------|--------|-----------|
| project_id    | 是    | String | 项目ID      |
| er_id         | 是    | String | 企业路由器实例ID |
| attachment_id | 是    | String | 连接ID      |

请求参数

无

响应参数

状态码：200

表 4-65 响应 Body 参数

| 参数         | 参数类型                      | 描述       |
|------------|---------------------------|----------|
| attachment | AttachmentResponse object | 连接基本信息详情 |
| request_id | String                    | 请求id     |

表 4-66 AttachmentResponse

| 参数          | 参数类型                 | 描述                                                                                                                      |
|-------------|----------------------|-------------------------------------------------------------------------------------------------------------------------|
| name        | String               | 连接名字                                                                                                                    |
| id          | String               | 连接ID                                                                                                                    |
| description | String               | 描述信息                                                                                                                    |
| state       | String               | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request freezed invalid |
| created_at  | String               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                        |
| updated_at  | String               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                        |
| tags        | Array of Tag objects | 企业路由器关联tag                                                                                                              |
| project_id  | String               | 项目ID                                                                                                                    |

| 参数                  | 参数类型   | 描述                                                                                                                                                                                                                                 |
|---------------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id         | String | 内部连接关联的资源ID                                                                                                                                                                                                                        |
| resource_type       | String | 内部连接关联的资源类型: <ul style="list-style-type: none"><li>• vgw: 云专线的虚拟网关</li><li>• vpn: vpn网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 (Peering)” 连接</li><li>• ecn: 企业连接网</li><li>• vpc: 虚拟私有云</li><li>• cfw: 云防火墙</li></ul> |
| resource_project_id | String | 资源所属项目ID                                                                                                                                                                                                                           |

表 4-67 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                                       |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键, 规则: <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格, 以及字符: 下划线“_”, 点“.”, 冒号“:”, 等号“=”, 加号“+”, 中划线“-”, “@”。</li><li>• _sys_开头属于系统标签, 租户不能输入。</li><li>• 标签key不能为空 (长度不能为0)。</li><li>• 长度在1-128之间。</li></ul> 最小长度: 1<br>最大长度: 128 |
| value | String | 标签值, 规则: <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格, 以及字符: 下划线“_”, 点“.”, 冒号“:”, 等号“=”, 加号“+”, 中划线“-”, “@”。</li><li>• 资源标签值可以为空 (empty or null)。</li><li>• 长度在0-255之间。</li></ul> 最小长度: 0<br>最大长度: 255                           |

## 请求示例

### 查询连接详情

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/212d3f0b-8496-472d-bc99-05a7c96d6655/attachments/b70aee08-c671-4cad-9fd5-7381d163bcc8
```

## 响应示例

### 状态码：200

OK

```
{
  "attachment" : {
    "id" : "b70aee08-c671-4cad-9fd5-7381d163bcc8",
    "name" : "vpc attachment",
    "description" : "vpc prod",
    "state" : "available",
    "created_at" : "2020-03-11T15:13:31Z",
    "updated_at" : "2020-03-11T15:13:31Z",
    "project_id" : "08d5a9564a704afda6039ae2babbe3c",
    "resource_id" : "b715e131-3371-4e17-a2de-4f669e24439a",
    "resource_type" : "vpc"
  },
  "request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ShowAttachmentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
```

```
        .build();
        ShowAttachmentRequest request = new ShowAttachmentRequest();
        request.withErId("{er_id}");
        request.withAttachmentId("{attachment_id}");
        try {
            ShowAttachmentResponse response = client.showAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowAttachmentRequest()
        request.er_id = "{er_id}"
        request.attachment_id = "{attachment_id}"
        response = client.show_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
```

```
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowAttachmentRequest{}
    request.ErId = "{er_id}"
    request.AttachmentId = "{attachment_id}"
    response, err := client.ShowAttachment(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.3.3 查询连接列表

### 功能介绍

查询企业路由器实例下的连接列表。

## 接口约束

- 支持过滤查询，过滤条件有state，resource\_type，resource\_id过滤条件可以重复和组合；
- 支持分页查询，limit和marker组合实现分页查询；
- 支持单字段排序，排序字段有[id,name,description,created\_at,updated\_at]，不支持多字段排序。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/attachments

表 4-68 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

表 4-69 Query 参数

| 参数     | 是否必选 | 参数类型             | 描述                                                                                                              |
|--------|------|------------------|-----------------------------------------------------------------------------------------------------------------|
| limit  | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                                                   |
| marker | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b>                                  |
| state  | 否    | Array of strings | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request invalid |

| 参数            | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                       |
|---------------|------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | 否    | Array of strings | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 (Peering)”连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| resource_id   | 否    | Array of strings | 连接对应的资源ID列表                                                                                                                                                                                              |
| sort_key      | 否    | Array of strings | 按关键字排序, 默认按照id排序, 可选值:id name state                                                                                                                                                                      |
| sort_dir      | 否    | Array of strings | 返回结果按照升序或降序排列, 默认为asc,降序为desc                                                                                                                                                                            |

请求参数

无

响应参数

状态码：200

表 4-70 响应 Body 参数

| 参数          | 参数类型                               | 描述      |
|-------------|------------------------------------|---------|
| attachments | Array of AttachmentDetails objects | 连接列表    |
| page_info   | PageInfo object                    | 分页查询的信息 |
| request_id  | String                             | 请求ID    |

表 4-71 AttachmentDetails

| 参数   | 参数类型   | 描述   |
|------|--------|------|
| id   | String | 连接ID |
| name | String | 连接名字 |

| 参数                  | 参数类型                        | 描述                                                                                                                                                                                                                                   |
|---------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| description         | String                      | 描述信息                                                                                                                                                                                                                                 |
| state               | String                      | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request frezed invalid                                                                                                               |
| created_at          | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                                                                     |
| updated_at          | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                                                                     |
| tags                | Array of <b>Tag</b> objects | 企业路由器关联tag                                                                                                                                                                                                                           |
| project_id          | String                      | 项目ID                                                                                                                                                                                                                                 |
| er_id               | String                      | Enterprise router ID                                                                                                                                                                                                                 |
| resource_id         | String                      | 内部连接关联的资源ID                                                                                                                                                                                                                          |
| resource_type       | String                      | 内部连接关联的资源类型: <ul style="list-style-type: none"><li>• vgw: 云专线的虚拟网关</li><li>• vpn: vpn网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>• ecn: 企业连接网</li><li>• vpc: 虚拟私有云</li><li>• cfw: 云防火墙</li></ul> |
| resource_project_id | String                      | 资源所属项目ID                                                                                                                                                                                                                             |
| associated          | Boolean                     | 表示此连接是否被关联                                                                                                                                                                                                                           |
| route_table_id      | String                      | 关联路由表id                                                                                                                                                                                                                              |

表 4-72 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

表 4-73 PageInfo

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| next_marker   | String  | 下一个marker标识，为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量             |

请求示例

查询连接列表

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/4ab54142-7c92-48ad-8288-77727a231052/attachments
```

响应示例

状态码：200

OK

```
{
  "attachments": [ {
    "id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "name": "vpc-001",
    "description": "vpc attachment",
    "resource_id": "b715e131-3371-4e17-a2de-4f669e24439a",
    "resource_type": "vpc",
    "project_id": "0605767a9980d5762fbcc00b3537e757",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  }, {
    "id": "6f83b848-8331-4271-ac0c-ef94b7686403",
    "name": "vpc-002",
    "description": "for test",
    "resource_id": "4b8567f6-358f-4a7f-8cd3-3cbb82c0b25f",
    "resource_type": "vpc",
    "project_id": "0605767a9980d5762fbcc00b3537e757",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  } ],
  "page_info": {
    "next_marker": "1",
    "current_count": 2
  },
  "request_id": "b715e131-3371-4e17-a2de-4f669e24439a"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListAttachmentsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(ErRegion.valueOf("<YOUR REGION>"))
        .build();
ListAttachmentsRequest request = new ListAttachmentsRequest();
request.withErId("{er_id}");
try {
    ListAttachmentsResponse response = client.listAttachments(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = ListAttachmentsRequest()
        request.er_id = "{er_id}"
        response = client.list_attachments(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
```

```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAttachmentsRequest{}
    request.ErId = "{er_id}"
    response, err := client.ListAttachments(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.3.4 接受连接创建

功能介绍

接受连接创建

调用方法

请参见[如何调用API](#)。

## URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/attachments/{attachment\_id}/accept

表 4-74 路径参数

| 参数            | 是否必选 | 参数类型   | 描述        |
|---------------|------|--------|-----------|
| project_id    | 是    | String | 项目ID      |
| er_id         | 是    | String | 企业路由器实例ID |
| attachment_id | 是    | String | 连接ID      |

## 请求参数

无

## 响应参数

状态码：202

表 4-75 响应 Body 参数

| 参数         | 参数类型                                      | 描述       |
|------------|-------------------------------------------|----------|
| attachment | <a href="#">AttachmentResponse</a> object | 连接基本信息详情 |
| request_id | String                                    | 请求id     |

表 4-76 AttachmentResponse

| 参数          | 参数类型   | 描述                                                                                                                      |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------|
| name        | String | 连接名字                                                                                                                    |
| id          | String | 连接ID                                                                                                                    |
| description | String | 描述信息                                                                                                                    |
| state       | String | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request freezed invalid |
| created_at  | String | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                        |

| 参数                  | 参数类型                                 | 描述                                                                                                                                                                                                                                   |
|---------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| updated_at          | String                               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                                                                     |
| tags                | Array of <a href="#">Tag</a> objects | 企业路由器关联tag                                                                                                                                                                                                                           |
| project_id          | String                               | 项目ID                                                                                                                                                                                                                                 |
| resource_id         | String                               | 内部连接关联的资源ID                                                                                                                                                                                                                          |
| resource_type       | String                               | 内部连接关联的资源类型: <ul style="list-style-type: none"><li>• vgw: 云专线的虚拟网关</li><li>• vpn: vpn网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>• ecn: 企业连接网</li><li>• vpc: 虚拟私有云</li><li>• cfw: 云防火墙</li></ul> |
| resource_project_id | String                               | 资源所属项目ID                                                                                                                                                                                                                             |

表 4-77 Tag

| 参数  | 参数类型   | 描述                                                                                                                                                                                                                                                               |
|-----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key | String | 标签键, 规则: <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格, 以及字符: 下划线 “_”, 点 “.”, 冒号 “:”, 等号 “=”, 加号 “+”, 中划线 “-”, “@”。</li><li>• _sys_开头属于系统标签, 租户不能输入。</li><li>• 标签key不能为空 ( 长度不能为0 )。</li><li>• 长度在1-128之间。</li></ul> 最小长度: 1<br>最大长度: 128 |

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                          |
|-------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| value | String | <p>标签值，规则：</p> <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> <p>最小长度：0<br/>最大长度：255</p> |

## 请求示例

接受创建连接

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/6f83b848-8331-4271-ac0c-ef94b7686402/attachments/6f83b848-8331-4271-ac0c-ef94b7686401/accept
```

## 响应示例

状态码：202

Accepted

```
{
  "attachment": {
    "id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "name": "vpc attachment",
    "description": "VPC连接",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z",
    "tags": null,
    "project_id": "0605767a9980d5762fbcc00b3537e757",
    "resource_id": "b715e131-3371-4e17-a2de-4f669e24439a",
    "resource_type": "VPC",
    "resource_project_id": "0605767a9980d5762fbcc00b3537e751"
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class AcceptAttachmentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        AcceptAttachmentRequest request = new AcceptAttachmentRequest();
        request.withErId("{er_id}");
        request.withAttachmentId("{attachment_id}");
        try {
            AcceptAttachmentResponse response = client.acceptAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
```

```
.with_credentials(credentials) \
.with_region(ErRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = AcceptAttachmentRequest()
    request.er_id = "{er_id}"
    request.attachment_id = "{attachment_id}"
    response = client.accept_attachment(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AcceptAttachmentRequest{
        request.ErId = "{er_id}"
        request.AttachmentId = "{attachment_id}"
    }
    response, err := client.AcceptAttachment(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.3.5 拒绝连接创建

### 功能介绍

拒绝连接创建

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/attachments/{attachment\_id}/  
reject

表 4-78 路径参数

| 参数            | 是否必选 | 参数类型   | 描述        |
|---------------|------|--------|-----------|
| project_id    | 是    | String | 项目ID      |
| er_id         | 是    | String | 企业路由器实例ID |
| attachment_id | 是    | String | 连接ID      |

### 请求参数

无

### 响应参数

状态码：200

表 4-79 响应 Body 参数

| 参数         | 参数类型                      | 描述       |
|------------|---------------------------|----------|
| attachment | AttachmentResponse object | 连接基本信息详情 |
| request_id | String                    | 请求id     |

表 4-80 AttachmentResponse

| 参数                  | 参数类型                 | 描述                                                                                                                                                                                                                                    |
|---------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name                | String               | 连接名字                                                                                                                                                                                                                                  |
| id                  | String               | 连接ID                                                                                                                                                                                                                                  |
| description         | String               | 描述信息                                                                                                                                                                                                                                  |
| state               | String               | 连接状态:pending available modifying deleting deleted failed pending_acceptance rejected initiating_request freezed invalid                                                                                                               |
| created_at          | String               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                                                                      |
| updated_at          | String               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                                                                      |
| tags                | Array of Tag objects | 企业路由器关联tag                                                                                                                                                                                                                            |
| project_id          | String               | 项目ID                                                                                                                                                                                                                                  |
| resource_id         | String               | 内部连接关联的资源ID                                                                                                                                                                                                                           |
| resource_type       | String               | 内部连接关联的资源类型: <ul style="list-style-type: none"><li>• vgw: 云专线的虚拟网关</li><li>• vpn: vpn网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering ) ” 连接</li><li>• ecn: 企业连接网</li><li>• vpc: 虚拟私有云</li><li>• cfw: 云防火墙</li></ul> |
| resource_project_id | String               | 资源所属项目ID                                                                                                                                                                                                                              |

表 4-81 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                   |
|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>_sys_ 开头属于系统标签，租户不能输入。</li><li>标签key不能为空（长度不能为0）。</li><li>长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>资源标签值可以为空（empty or null）。</li><li>长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                         |

## 请求示例

### 拒绝创建连接

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/6f83b848-8331-4271-ac0c-ef94b7686402/attachments/6f83b848-8331-4271-ac0c-ef94b7686401/reject
```

## 响应示例

### 状态码：200

OK

```
{
  "attachment": {
    "id": "6f83b848-8331-4271-ac0c-ef94b7686402,",
    "name": "vpc attachment,",
    "description": "VPC连接,",
    "state": "available",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z",
    "tags": null,
    "project_id": "0605767a9980d5762fbcc00b3537e757",
```

```
"resource_id" : "b715e131-3371-4e17-a2de-4f669e24439a",
"resource_type" : "VPC",
"resource_project_id" : "0605767a9980d5762fbcc00b3537e751"
},
"request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class RejectAttachmentSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        RejectAttachmentRequest request = new RejectAttachmentRequest();
        request.withErId("{er_id}");
        request.withAttachmentId("{attachment_id}");
        try {
            RejectAttachmentResponse response = client.rejectAttachment(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = RejectAttachmentRequest()
        request.er_id = "{er_id}"
        request.attachment_id = "{attachment_id}"
        response = client.reject_attachment(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).
WithCredential(auth).
Build())

request := &model.RejectAttachmentRequest{}
request.ErId = "{er_id}"
request.AttachmentId = "{attachment_id}"
response, err := client.RejectAttachment(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.4 路由表

4.4.1 创建路由表

功能介绍

路由表是企业路由器收发报文的依据，包含了连接的关联关系，传播关系以及路由信息。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/route-tables

表 4-82 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

| 参数    | 是否必选 | 参数类型   | 描述        |
|-------|------|--------|-----------|
| er_id | 是    | String | 企业路由器实例ID |

请求参数

表 4-83 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-84 请求 Body 参数

| 参数          | 是否必选 | 参数类型                    | 描述  |
|-------------|------|-------------------------|-----|
| route_table | 否    | CreateRouteTable object | 路由表 |

表 4-85 CreateRouteTable

| 参数          | 是否必选 | 参数类型                 | 描述                                                                      |
|-------------|------|----------------------|-------------------------------------------------------------------------|
| name        | 是    | String               | 路由器表名称，取值范围：1-64 个字符，支持数字、字母、中文、_(下划线)、-(中划线)、.(点)<br>最小长度：1<br>最大长度：64 |
| description | 否    | String               | 路由器表描述信息<br>最小长度：0<br>最大长度：255                                          |
| tags        | 否    | Array of Tag objects | 标签信息                                                                    |

表 4-86 Tag

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                           |
|-------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | 否    | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | 否    | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

响应参数

状态码：202

表 4-87 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-88 响应 Body 参数

| 参数          | 参数类型                              | 描述   |
|-------------|-----------------------------------|------|
| route_table | <a href="#">RouteTable</a> object | 路由表  |
| request_id  | String                            | 请求ID |

表 4-89 RouteTable

| 参数                     | 参数类型                                 | 描述                                                            |
|------------------------|--------------------------------------|---------------------------------------------------------------|
| id                     | String                               | 路由表的id                                                        |
| name                   | String                               | 路由表名字                                                         |
| description            | String                               | 描述信息                                                          |
| is_default_association | Boolean                              | 是否为默认关联的路由表<br>缺省值: <b>false</b>                              |
| is_default_propagation | Boolean                              | 是否为默认传递路由表                                                    |
| state                  | String                               | 路由表状态，支持的状态有pending   available   deleting   deleted   failed |
| tags                   | Array of <a href="#">Tag</a> objects | 标签                                                            |
| created_at             | String                               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                              |
| updated_at             | String                               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                              |

表 4-90 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

请求示例

在企业路由器内创建路由表，名称为“my-route-table”

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/915a14a6-867b-4af7-83d1-70efceb146f0/route-tables

{
  "route_table": {
    "name": "my-route-table",
    "tags": [ {
      "key": "key",
      "value": "value"
    } ]
  }
}
```

响应示例

状态码：202

Accepted

```
{
  "route_table": {
```

```
{
  "id": "4ab54142-7c92-48ad-8288-77727a231052",
  "name": "my-route-table",
  "is_default_association": false,
  "is_default_propagation": false,
  "state": "pending",
  "created_at": "2020-03-11T15:13:31Z",
  "updated_at": "2020-03-11T15:13:31Z",
  "tags": [ {
    "key": "key",
    "value": "value"
  } ]
},
"request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

在企业路由器内创建路由表，名称为“my-route-table”

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateRouteTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateRouteTableRequest request = new CreateRouteTableRequest();
        request.withErId("{er_id}");
        CreateRouteTableRequestBody body = new CreateRouteTableRequestBody();
        List<Tag> listRouteTableTags = new ArrayList<>();
        listRouteTableTags.add(
            new Tag()
                .withKey("key")
                .withValue("value")
        );
    }
}
```

```
CreateRouteTable routeTablebody = new CreateRouteTable();
routeTablebody.setName("my-route-table")
    .withTags(listRouteTableTags);
body.withRouteTable(routeTablebody);
request.withBody(body);
try {
    CreateRouteTableResponse response = client.createRouteTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

在企业路由器内创建路由表，名称为 “my-route-table”

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateRouteTableRequest()
        request.er_id = "{er_id}"
        listTagsRouteTable = [
            Tag(
                key="key",
                value="value"
            )
        ]
        routeTablebody = CreateRouteTable(
            name="my-route-table",
            tags=listTagsRouteTable
        )
        request.body = CreateRouteTableRequestBody(
            route_table=routeTablebody
        )
        response = client.create_route_table(request)
        print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

在企业路由器内创建路由表，名称为 “my-route-table”

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateRouteTableRequest{}
    request.ErId = "{er_id}"
    keyTags := "key"
    valueTags := "value"
    var listTagsRouteTable = []model.Tag{
        {
            Key: &keyTags,
            Value: &valueTags,
        },
    }
    routeTablebody := &model.CreateRouteTable{
        Name: "my-route-table",
        Tags: &listTagsRouteTable,
    }
    request.Body = &model.CreateRouteTableRequestBody{
        RouteTable: routeTablebody,
    }
    response, err := client.CreateRouteTable(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.4.2 更新路由表信息

### 功能介绍

更新路由表基本信息，如名称，描述等。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}

表 4-91 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

### 请求参数

表 4-92 请求 Body 参数

| 参数          | 是否必选 | 参数类型                                    | 描述       |
|-------------|------|-----------------------------------------|----------|
| route_table | 否    | <a href="#">UpdateRouteTable</a> object | 更新路由表请求体 |

表 4-93 UpdateRouteTable

| 参数          | 是否必选 | 参数类型   | 描述                                                                      |
|-------------|------|--------|-------------------------------------------------------------------------|
| name        | 否    | String | 路由器表名称，取值范围：1-64 个字符，支持数字、字母、中文、_(下划线)、-（中划线）、.（点）<br>最小长度：1<br>最大长度：64 |
| description | 否    | String | 路由器表描述信息<br>最小长度：0<br>最大长度：255                                          |

响应参数

状态码：200

表 4-94 响应 Body 参数

| 参数          | 参数类型              | 描述   |
|-------------|-------------------|------|
| route_table | RouteTable object | 路由表  |
| request_id  | String            | 请求ID |

表 4-95 RouteTable

| 参数                     | 参数类型                 | 描述                                                            |
|------------------------|----------------------|---------------------------------------------------------------|
| id                     | String               | 路由表的id                                                        |
| name                   | String               | 路由表名字                                                         |
| description            | String               | 描述信息                                                          |
| is_default_association | Boolean              | 是否为默认关联的路由表<br>缺省值：false                                      |
| is_default_propagation | Boolean              | 是否为默认传递路由表                                                    |
| state                  | String               | 路由表状态，支持的状态有pending   available   deleting   deleted   failed |
| tags                   | Array of Tag objects | 标签                                                            |

| 参数         | 参数类型   | 描述                               |
|------------|--------|----------------------------------|
| created_at | String | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| updated_at | String | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |

表 4-96 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

请求示例

更新企业路由器内路由表的信息，修改路由表名称为“new-rtb”

```
PUT https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/915a14a6-867b-4af7-83d1-70efceb146f5/route-tables/4ab54142-7c92-48ad-8288-77727a231052
{
  "route_table": {
    "name": "new-rtb"
  }
}
```

## 响应示例

状态码：200

OK

```
{
  "route_table" : {
    "id" : "4ab54142-7c92-48ad-8288-77727a231052",
    "name" : "new-rtb",
    "is_default_association" : false,
    "is_default_propagation" : false,
    "state" : "pending",
    "created_at" : "2020-03-11T15:13:31Z",
    "updated_at" : "2020-03-11T15:13:31Z"
  },
  "request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新企业路由器内路由表的信息，修改路由表名称为“new-rtb”

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class UpdateRouteTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        UpdateRouteTableRequest request = new UpdateRouteTableRequest();
        request.withErId("{er_id}");
        request.withRouteTableId("{route_table_id}");
        UpdateRouteTableRequestBody body = new UpdateRouteTableRequestBody();
        UpdateRouteTable routeTablebody = new UpdateRouteTable();
        routeTablebody.withName("new-rtb");
        body.withRouteTable(routeTablebody);
    }
}
```

```
request.withBody(body);
try {
    UpdateRouteTableResponse response = client.updateRouteTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

更新企业路由器内路由表的信息，修改路由表名称为 “new-rtb”

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateRouteTableRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        routeTablebody = UpdateRouteTable(
            name="new-rtb"
        )
        request.body = UpdateRouteTableRequestBody(
            route_table=routeTablebody
        )
        response = client.update_route_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

更新企业路由器内路由表的信息，修改路由表名称为 “new-rtb”

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.NewErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateRouteTableRequest{
        request.ErId = "{er_id}"
        request.RouteTableId = "{route_table_id}"
        nameRouteTable := "new-rtb"
        routeTablebody := &model.UpdateRouteTable{
            Name: &nameRouteTable,
        }
        request.Body = &model.UpdateRouteTableRequestBody{
            RouteTable: routeTablebody,
        }
        response, err := client.UpdateRouteTable(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

### 4.4.3 查询路由表详情

#### 功能介绍

查询路由表详情

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}

表 4-97 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

#### 请求参数

无

#### 响应参数

状态码：200

表 4-98 响应 Body 参数

| 参数          | 参数类型                              | 描述   |
|-------------|-----------------------------------|------|
| route_table | <a href="#">RouteTable</a> object | 路由表  |
| request_id  | String                            | 请求ID |

表 4-99 RouteTable

| 参数          | 参数类型   | 描述     |
|-------------|--------|--------|
| id          | String | 路由表的id |
| name        | String | 路由表名字  |
| description | String | 描述信息   |

| 参数                     | 参数类型                        | 描述                                                             |
|------------------------|-----------------------------|----------------------------------------------------------------|
| is_default_association | Boolean                     | 是否为默认关联的路由表<br>缺省值: <b>false</b>                               |
| is_default_propagation | Boolean                     | 是否为默认传递路由表                                                     |
| state                  | String                      | 路由表状态, 支持的状态有pending   available   deleting   deleted   failed |
| tags                   | Array of <b>Tag</b> objects | 标签                                                             |
| created_at             | String                      | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                               |
| updated_at             | String                      | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                               |

表 4-100 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                                                            |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键, 规则: <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格, 以及字符: 下划线 “_”, 点 “.”, 冒号 “:”, 等号 “=”, 加号 “+”, 中划线 “-”, “@”。</li><li>• _sys_ 开头属于系统标签, 租户不能输入。</li><li>• 标签key不能为空 (长度不能为0)。</li><li>• 长度在1-128之间。</li></ul> 最小长度: <b>1</b><br>最大长度: <b>128</b> |
| value | String | 标签值, 规则: <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格, 以及字符: 下划线 “_”, 点 “.”, 冒号 “:”, 等号 “=”, 加号 “+”, 中划线 “-”, “@”。</li><li>• 资源标签值可以为空 (empty or null)。</li><li>• 长度在0-255之间。</li></ul> 最小长度: <b>0</b><br>最大长度: <b>255</b>                            |

## 请求示例

### 查询路由表详情

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/  
915a14a6-867b-4af7-83d1-70efceb146f6/route-tables/4ab54142-7c92-48ad-8288-77727a231052
```

## 响应示例

### 状态码：200

OK

```
{  
  "route_table": {  
    "id": "4ab54142-7c92-48ad-8288-77727a231052",  
    "name": "my-route-table",  
    "description": "rtb-for-a",  
    "is_default_association": false,  
    "is_default_propagation": false,  
    "state": "available",  
    "created_at": "2020-03-11T15:13:31Z",  
    "updated_at": "2020-03-11T15:13:31Z"  
  },  
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.er.v3.region.ErRegion;  
import com.huaweicloud.sdk.er.v3.*;  
import com.huaweicloud.sdk.er.v3.model.*;  
  
public class ShowRouteTableSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        ErClient client = ErClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))  
            .build();  
    }  
}
```

```
ShowRouteTableRequest request = new ShowRouteTableRequest();
request.withErId("{er_id}");
request.withRouteTableId("{route_table_id}");
try {
    ShowRouteTableResponse response = client.showRouteTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowRouteTableRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        response = client.show_route_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
```

```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowRouteTableRequest{
        request.ErId = "{er_id}"
        request.RouteTableId = "{route_table_id}"
    }
    response, err := client.ShowRouteTable(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.4.4 查询路由表列表

功能介绍

查询路由表列表。

接口约束

- 支持分页查询, 支持过滤查询: state, is\_default\_propagation\_route\_table, is\_default\_association\_route\_table。

- 支持单字段排序，排序字段有[id,name,description,created\_at,updated\_at]，不支持多字段排序。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/route-tables

表 4-101 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

表 4-102 Query 参数

| 参数                           | 是否必选 | 参数类型             | 描述                                                                             |
|------------------------------|------|------------------|--------------------------------------------------------------------------------|
| limit                        | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                  |
| marker                       | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b> |
| state                        | 否    | Array of strings | 状态                                                                             |
| is_default_propagation_table | 否    | Boolean          | 是否为默认传播路由表                                                                     |
| is_default_association_table | 否    | Boolean          | 是否为默认关联路由表                                                                     |
| sort_key                     | 否    | Array of strings | 按关键字排序，默认按照id排序，可选值:id name state                                              |
| sort_dir                     | 否    | Array of strings | 返回结果按照升序或降序排列，默认为asc,降序为desc                                                   |

请求参数

无

响应参数

状态码：200

表 4-103 响应 Body 参数

| 参数           | 参数类型                                        | 描述      |
|--------------|---------------------------------------------|---------|
| route_tables | Array of <a href="#">RouteTable</a> objects | 路由表列表   |
| request_id   | String                                      | 请求ID    |
| page_info    | <a href="#">PageInfo</a> object             | 分页查询的信息 |

表 4-104 RouteTable

| 参数                     | 参数类型                                 | 描述                                                            |
|------------------------|--------------------------------------|---------------------------------------------------------------|
| id                     | String                               | 路由表的id                                                        |
| name                   | String                               | 路由表名字                                                         |
| description            | String                               | 描述信息                                                          |
| is_default_association | Boolean                              | 是否为默认关联的路由表<br>缺省值： <b>false</b>                              |
| is_default_propagation | Boolean                              | 是否为默认传递路由表                                                    |
| state                  | String                               | 路由表状态，支持的状态有pending   available   deleting   deleted   failed |
| tags                   | Array of <a href="#">Tag</a> objects | 标签                                                            |
| created_at             | String                               | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                              |
| updated_at             | String                               | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                              |

表 4-105 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                           |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

表 4-106 PageInfo

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| next_marker   | String  | 下一个marker标识，为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量             |

请求示例

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/915a14a6-867b-4af7-83d1-70efceb146f5/route-tables
```

响应示例

状态码：200  
OK

```
{
  "route_tables" : [ {
    "id" : "4ab54142-7c92-48ad-8288-77727a231052",
    "is_default_association" : false,
    "is_default_propagation" : false,
    "name" : "my-router-table1",
    "description" : "rtb-for-a",
    "state" : "available",
    "tags" : [ {
      "key" : "key",
      "value" : "value"
    } ]
  }, {
    "id" : "4ab54142-7c92-48ad-8288-77727a231053",
    "is_default_association" : false,
    "is_default_propagation" : false,
    "name" : "my-router-table2",
    "description" : "rtb-for-b",
    "state" : "available",
    "tags" : [ {
      "key" : "key",
      "value" : "value"
    } ]
  } ],
  "page_info" : {
    "next_marker" : "1",
    "current_count" : 2
  },
  "request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListRouteTablesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(ErRegion.valueOf("<YOUR REGION>"))
        .build();
ListRouteTablesRequest request = new ListRouteTablesRequest();
request.withErId("{er_id}");
try {
    ListRouteTablesResponse response = client.listRouteTables(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = ListRouteTablesRequest()
        request.er_id = "{er_id}"
        response = client.list_route_tables(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
```

```
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListRouteTablesRequest{}
    request.ErId = "{er_id}"
    response, err := client.ListRouteTables(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.4.5 删除路由表

功能介绍

删除路由表

调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}

表 4-107 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

## 请求参数

无

## 响应参数

无

## 请求示例

```
DELETE https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/4ab54142-7c92-48ad-8288-77727a231056/route-tables/4ab54142-7c92-48ad-8288-77727a231052
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DeleteRouteTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
DeleteRouteTableRequest request = new DeleteRouteTableRequest();
request.withErId("{er_id}");
request.withRouteTableId("{route_table_id}");
try {
    DeleteRouteTableResponse response = client.deleteRouteTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteRouteTableRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        response = client.delete_route_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteRouteTableRequest{}
    request.ErId = "{er_id}"
    request.RouteTableId = "{route_table_id}"
    response, err := client.DeleteRouteTable(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.5 关联

### 4.5.1 创建路由关联

#### 功能介绍

每个连接只能关联到一张路由表。通过创建关联将连接关联到路由表，从该连接收到的报文会用被关联的路由表进行路由。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}/associate

表 4-108 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

#### 请求参数

表 4-109 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-110 请求 Body 参数

| 参数            | 是否必选 | 参数类型   | 描述                |
|---------------|------|--------|-------------------|
| attachment_id | 否    | String | 连接唯一标识<br>最大长度：36 |

| 参数           | 是否必选 | 参数类型                             | 描述      |
|--------------|------|----------------------------------|---------|
| route_policy | 否    | <b>ExportRoute Policy</b> object | 出方向路由策略 |

表 4-111 ExportRoutePolicy

| 参数               | 是否必选 | 参数类型   | 描述                                   |
|------------------|------|--------|--------------------------------------|
| export_policy_id | 否    | String | 出方向Ipv4协议路由策略id<br>最小长度：0<br>最大长度：36 |

## 响应参数

状态码：202

表 4-112 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-113 响应 Body 参数

| 参数          | 参数类型                      | 描述   |
|-------------|---------------------------|------|
| association | <b>Association</b> object | 关联详情 |
| request_id  | String                    | 请求ID |

表 4-114 Association

| 参数             | 参数类型   | 描述                 |
|----------------|--------|--------------------|
| id             | String | 关联唯一标识<br>最大长度：36  |
| route_table_id | String | 路由表唯一标识<br>最大长度：36 |
| attachment_id  | String | 连接唯一标识<br>最大长度：36  |

| 参数            | 参数类型                            | 描述                                                                                                                                                                                                          |
|---------------|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | String                          | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| resource_id   | String                          | 连接的资源唯一标识                                                                                                                                                                                                   |
| state         | String                          | 状态:pending available modifying deleting deleted failed                                                                                                                                                      |
| created_at    | String                          | 资源创建时间<br>采用UTC时间<br>格式: YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                           |
| updated_at    | String                          | 资源更新时间<br>采用UTC时间<br>格式: YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                           |
| route_policy  | <b>ExportRoutePolicy</b> object | 出方向路由策略                                                                                                                                                                                                     |

表 4-115 ExportRoutePolicy

| 参数               | 参数类型   | 描述                                     |
|------------------|--------|----------------------------------------|
| export_policy_id | String | 出方向Ipv4协议路由策略id<br>最小长度: 0<br>最大长度: 36 |

请求示例

在企业路由器的路由表中创建关联, 将指定连接关联至路由表中

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/915a14a6-867b-4af7-83d1-70efceb146f8/route-tables/915a14a6-867b-4af7-83d1-70efceb146f9/associate
{
  "attachment_id": "a5b7d209-dc02-4c46-a51f-805eadd3de64"
}
```

响应示例

状态码: 202

## Accepted

```
{
  "association": {
    "id": "915a14a6-867b-4af7-83d1-70efceb146f9",
    "route_table_id": "91c013e2-d65a-474e-9177-c3e1799ca726",
    "attachment_id": "a5b7d209-dc02-4c46-a51f-805eadd3de64",
    "resource_type": "vpc",
    "resource_id": "4e5fe97c-82bc-432e-87d8-06d7e157dffa",
    "state": "pending",
    "created_at": "2020-03-11T15:13:31",
    "updated_at": "2020-03-11T15:13:31",
    "route_policy": {
      "export_policy_id": ""
    }
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f8"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

在企业路由器的路由表中创建关联，将指定连接关联至路由表中

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class AssociateRouteTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        AssociateRouteTableRequest request = new AssociateRouteTableRequest();
        request.withErId("{er_id}");
        request.withRouteTableId("{route_table_id}");
        AssociationRequestBody body = new AssociationRequestBody();
        body.withAttachmentId("a5b7d209-dc02-4c46-a51f-805eadd3de64");
        request.withBody(body);
        try {
```

```
        AssociateRouteTableResponse response = client.associateRouteTable(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

在企业路由器的路由表中创建关联，将指定连接关联至路由表中

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = AssociateRouteTableRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        request.body = AssociationRequestBody(
            attachment_id="a5b7d209-dc02-4c46-a51f-805eadd3de64"
        )
        response = client.associate_route_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

在企业路由器的路由表中创建关联，将指定连接关联至路由表中

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AssociateRouteTableRequest{}
    request.ErId = "{er_id}"
    request.RouteTableId = "{route_table_id}"
    attachmentIdAssociationRequestBody := "a5b7d209-dc02-4c46-a51f-805eadd3de64"
    request.Body = &model.AssociationRequestBody{
        AttachmentId: &attachmentIdAssociationRequestBody,
    }
    response, err := client.AssociateRouteTable(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

错误码

请参见[错误码](#)。

## 4.5.2 查询路由关联列表

### 功能介绍

查询路由关联列表。

### 接口约束

- 支持分页查询, 支持过滤查询: state, resource\_type, attachment\_id。
- 支持单字段排序, 排序字段有[id,created\_at,updated\_at], 不支持多字段排序。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}/associations

表 4-116 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

表 4-117 Query 参数

| 参数            | 是否必选 | 参数类型             | 描述                                                                               |
|---------------|------|------------------|----------------------------------------------------------------------------------|
| limit         | 否    | Integer          | 每页返回的个数。<br>取值范围: 0~2000。<br>最小值: <b>0</b><br>最大值: <b>2000</b>                   |
| marker        | 否    | String           | 查询的页码id, 为空时为查询第一页。<br>使用说明: 必须与limit一起使用。<br>最小长度: <b>1</b><br>最大长度: <b>128</b> |
| attachment_id | 否    | Array of strings | 连接ID                                                                             |

| 参数            | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                       |
|---------------|------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | 否    | Array of strings | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 (Peering)”连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| state         | 否    | Array of strings | 状态:pending available modifying deleting deleted failed                                                                                                                                                   |
| sort_key      | 否    | Array of strings | 按关键字排序, 默认按照id排序, 可选值:id name state                                                                                                                                                                      |
| sort_dir      | 否    | Array of strings | 返回结果按照升序或降序排列, 默认为asc,降序为desc                                                                                                                                                                            |

请求参数

无

响应参数

状态码：200

表 4-118 响应 Body 参数

| 参数           | 参数类型                                         | 描述                       |
|--------------|----------------------------------------------|--------------------------|
| associations | Array of <a href="#">Association</a> objects | 路由表关联列表<br>数组长度：0 - 2000 |
| request_id   | String                                       | 请求ID                     |
| page_info    | <a href="#">PageInfo</a> object              | 分页查询的信息                  |

表 4-119 Association

| 参数 | 参数类型   | 描述                |
|----|--------|-------------------|
| id | String | 关联唯一标识<br>最大长度：36 |

| 参数             | 参数类型                                     | 描述                                                                                                                                                                                               |
|----------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| route_table_id | String                                   | 路由表唯一标识<br>最大长度：36                                                                                                                                                                               |
| attachment_id  | String                                   | 连接唯一标识<br>最大长度：36                                                                                                                                                                                |
| resource_type  | String                                   | <ul style="list-style-type: none"><li>vpc：虚拟私有云</li><li>vpn：vpn网关</li><li>vgw：云专线的虚拟网关</li><li>peering：对等连接，通过云连接CC加载不同区域的企业路由器来创建“对等连接（Peering）”连接</li><li>ecn：企业连接网</li><li>cfw：云防火墙</li></ul> |
| resource_id    | String                                   | 连接的资源唯一标识                                                                                                                                                                                        |
| state          | String                                   | 状态:pending available modifying deleting deleted failed                                                                                                                                           |
| created_at     | String                                   | 资源创建时间<br>采用UTC时间<br>格式：YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                 |
| updated_at     | String                                   | 资源更新时间<br>采用UTC时间<br>格式：YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                 |
| route_policy   | <a href="#">ExportRoutePolicy</a> object | 出方向路由策略                                                                                                                                                                                          |

表 4-120 ExportRoutePolicy

| 参数               | 参数类型   | 描述                                   |
|------------------|--------|--------------------------------------|
| export_policy_id | String | 出方向Ipv4协议路由策略id<br>最小长度：0<br>最大长度：36 |

表 4-121 PageInfo

| 参数          | 参数类型   | 描述                    |
|-------------|--------|-----------------------|
| next_marker | String | 下一个marker标识，为空时表示最后一页 |

| 参数            | 参数类型    | 描述        |
|---------------|---------|-----------|
| current_count | Integer | 当前列表中资源数量 |

## 请求示例

查询路由表的关联信息

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/915a14a6-867b-4af7-83d1-70efceb146f5/route-tables/915a14a6-867b-4af7-83d1-70efceb146f9/associations
```

## 响应示例

状态码：200

OK

```
{
  "associations" : [ {
    "id" : "915a14a6-867b-4af7-83d1-70efceb146f9",
    "route_table_id" : "91c013e2-d65a-474e-9177-c3e1799ca726",
    "attachment_id" : "a5b7d209-dc02-4c46-a51f-805eadd3de64",
    "resource_type" : "vpc",
    "resource_id" : "4e5fe97c-82bc-432e-87d8-06d7e157dffa",
    "state" : "pending",
    "created_at" : "2020-03-11T15:13:31",
    "updated_at" : "2020-03-11T15:13:31",
    "route_policy" : {
      "export_policy_id" : ""
    }
  } ],
  "page_info" : {
    "next_marker" : "1",
    "current_count" : 1
  },
  "request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListAssociationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
    }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
ListAssociationsRequest request = new ListAssociationsRequest();
request.withErId("{er_id}");
request.withRouteTableId("{route_table_id}");
try {
    ListAssociationsResponse response = client.listAssociations(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAssociationsRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        response = client.list_associations(request)
        print(response)
    except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAssociationsRequest{}
    request.ErId = "{er_id}"
    request.RouteTableId = "{route_table_id}"
    response, err := client.ListAssociations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.5.3 删除路由关联

### 功能介绍

解绑连接和路由表的关联关系。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}/disassociate

表 4-122 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

### 请求参数

表 4-123 请求 Body 参数

| 参数            | 是否必选 | 参数类型                                     | 描述                |
|---------------|------|------------------------------------------|-------------------|
| attachment_id | 否    | String                                   | 连接唯一标识<br>最大长度：36 |
| route_policy  | 否    | <a href="#">ExportRoutePolicy</a> object | 出方向路由策略           |

表 4-124 ExportRoutePolicy

| 参数               | 是否必选 | 参数类型   | 描述                                   |
|------------------|------|--------|--------------------------------------|
| export_policy_id | 否    | String | 出方向Ipv4协议路由策略id<br>最小长度：0<br>最大长度：36 |

## 响应参数

无

## 请求示例

删除路由表关联

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/915a14a6-867b-4af7-83d1-70efceb146f8/route-tables/915a14a6-867b-4af7-83d1-70efceb146f9/disassociate
{
  "attachment_id" : "2609ec92-ac19-4467-a527-7f9e690e7836"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除路由表关联

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DisassociateRouteTableSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DisassociateRouteTableRequest request = new DisassociateRouteTableRequest();
        request.withErId("{er_id}");
        request.withRouteTableId("{route_table_id}");
        AssociationRequestBody body = new AssociationRequestBody();
```

```
body.withAttachmentId("2609ec92-ac19-4467-a527-7f9e690e7836");
request.withBody(body);
try {
    DisassociateRouteTableResponse response = client.disassociateRouteTable(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 删除路由表关联

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisassociateRouteTableRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        request.body = AssociationRequestBody(
            attachment_id="2609ec92-ac19-4467-a527-7f9e690e7836"
        )
        response = client.disassociate_route_table(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 删除路由表关联

```
package main
```

```
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := er.NewErClient(  
        er.ErClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DisassociateRouteTableRequest{  
        request.ErId = "{er_id}"  
        request.RouteTableId = "{route_table_id}"  
        attachmentIdAssociationRequestBody:= "2609ec92-ac19-4467-a527-7f9e690e7836"  
        request.Body = &model.AssociationRequestBody{  
            AttachmentId: &attachmentIdAssociationRequestBody,  
        }  
        response, err := client.DisassociateRouteTable(request)  
        if err == nil {  
            fmt.Printf("%v\n", response)  
        } else {  
            fmt.Println(err)  
        }  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.6 传播

### 4.6.1 创建路由传播

#### 功能介绍

每个连接可以和多个路由表建立传播关系，从该连接学习到的路由会应用到具有传播关系的路由表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}/enable-propagations

表 4-125 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

#### 请求参数

表 4-126 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-127 请求 Body 参数

| 参数            | 是否必选 | 参数类型   | 描述     |
|---------------|------|--------|--------|
| attachment_id | 否    | String | 连接唯一标识 |

| 参数           | 是否必选 | 参数类型                             | 描述      |
|--------------|------|----------------------------------|---------|
| route_policy | 否    | <b>ImportRoute Policy</b> object | 入方向路由策略 |

表 4-128 ImportRoutePolicy

| 参数               | 是否必选 | 参数类型   | 描述                                   |
|------------------|------|--------|--------------------------------------|
| import_policy_id | 否    | String | 入方向Ipv4协议路由策略id<br>最小长度：0<br>最大长度：36 |

响应参数

状态码：202

表 4-129 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-130 响应 Body 参数

| 参数          | 参数类型                      | 描述   |
|-------------|---------------------------|------|
| propagation | <b>Propagation</b> object | 传播详情 |
| request_id  | String                    | 请求ID |

表 4-131 Propagation

| 参数             | 参数类型   | 描述      |
|----------------|--------|---------|
| id             | String | 关联唯一标识  |
| project_id     | String | 项目ID    |
| er_id          | String | 企业路由器id |
| route_table_id | String | 路由表唯一标识 |
| attachment_id  | String | 连接唯一标识  |

| 参数            | 参数类型                     | 描述                                                                                                                                                                                                          |
|---------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | String                   | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| resource_id   | String                   | 连接的资源唯一标识                                                                                                                                                                                                   |
| route_policy  | ImportRoutePolicy object | 入方向路由策略                                                                                                                                                                                                     |
| state         | String                   | 状态:pending available modifying deleting deleted failed                                                                                                                                                      |
| created_at    | String                   | 资源创建时间<br>采用UTC时间<br>格式: YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                           |
| updated_at    | String                   | 资源更新时间<br>采用UTC时间<br>格式: YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                           |

表 4-132 ImportRoutePolicy

| 参数               | 参数类型   | 描述                                     |
|------------------|--------|----------------------------------------|
| import_policy_id | String | 入方向Ipv4协议路由策略id<br>最小长度: 0<br>最大长度: 36 |

请求示例

在企业路由器的路由表中，为指定连接创建传播，学习连接的路由信息

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/ad838a5e-  
dbef-22cc-b1d9-cb46bef77ae8/route-tables/915a14a6-867b-4af7-83d1-70efceb146f9/enable-propagations  
  
{  
  "attachment_id": "a5b7d209-dc02-4c46-a51f-805eadd3de64"  
}
```

响应示例

状态码: 202

## Accepted

```
{
  "propagation" : {
    "id" : "915a14a6-867b-4af7-83d1-70efceb146f9",
    "route_table_id" : "91c013e2-d65a-474e-9177-c3e1799ca726",
    "attachment_id" : "a5b7d209-dc02-4c46-a51f-805eadd3de64",
    "resource_type" : "vpc",
    "resource_id" : "4e5fe97c-82bc-432e-87d8-06d7e157dffa",
    "route_policy" : {
      "import_policy_id" : ""
    },
    "state" : "pending",
    "created_at" : "2020-03-11T15:13:31Z",
    "updated_at" : "2020-03-11T15:13:31Z"
  },
  "request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

在企业路由器的路由表中，为指定连接创建传播，学习连接的路由信息

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class EnablePropagationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        EnablePropagationRequest request = new EnablePropagationRequest();
        request.withErId("{er_id}");
        request.withRouteTableId("{route_table_id}");
        PropagationRequestBody body = new PropagationRequestBody();
        body.withAttachmentId("a5b7d209-dc02-4c46-a51f-805eadd3de64");
        request.withBody(body);
        try {
```

```
        EnablePropagationResponse response = client.enablePropagation(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

在企业路由器的路由表中，为指定连接创建传播，学习连接的路由信息

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = EnablePropagationRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        request.body = PropagationRequestBody(
            attachment_id="a5b7d209-dc02-4c46-a51f-805eadd3de64"
        )
        response = client.enable_propagation(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

在企业路由器的路由表中，为指定连接创建传播，学习连接的路由信息

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.EnablePropagationRequest{}
    request.ErId = "{er_id}"
    request.RouteTableId = "{route_table_id}"
    attachmentIdPropagationRequestBody := "a5b7d209-dc02-4c46-a51f-805eadd3de64"
    request.Body = &model.PropagationRequestBody{
        AttachmentId: &attachmentIdPropagationRequestBody,
    }
    response, err := client.EnablePropagation(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

错误码

请参见[错误码](#)。

## 4.6.2 查询路由传播列表

### 功能介绍

查询路由传播列表。

### 接口约束

- 支持分页查询, 支持过滤查询: state, resource\_type, attachment\_id。
- 支持单字段排序, 排序字段有[id,created\_at,updated\_at], 不支持多字段排序。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}/propagations

表 4-133 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

表 4-134 Query 参数

| 参数            | 是否必选 | 参数类型             | 描述                                                                               |
|---------------|------|------------------|----------------------------------------------------------------------------------|
| limit         | 否    | Integer          | 每页返回的个数。<br>取值范围: 0~2000。<br>最小值: <b>0</b><br>最大值: <b>2000</b>                   |
| marker        | 否    | String           | 查询的页码id, 为空时为查询第一页。<br>使用说明: 必须与limit一起使用。<br>最小长度: <b>1</b><br>最大长度: <b>128</b> |
| attachment_id | 否    | Array of strings | 连接ID                                                                             |

| 参数            | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                       |
|---------------|------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | 否    | Array of strings | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 (Peering)”连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| state         | 否    | Array of strings | 资源状态:pending available modifying deleting deleted failed freezed                                                                                                                                         |
| sort_key      | 否    | Array of strings | 按关键字排序, 默认按照id排序, 可选值:id name state                                                                                                                                                                      |
| sort_dir      | 否    | Array of strings | 返回结果按照升序或降序排列, 默认为asc,降序为desc                                                                                                                                                                            |

请求参数

无

响应参数

状态码：200

表 4-135 响应 Body 参数

| 参数           | 参数类型                                | 描述                      |
|--------------|-------------------------------------|-------------------------|
| propagations | Array of <b>Propagation</b> objects | 路由传播列表<br>数组长度：0 - 2000 |
| request_id   | String                              | 请求ID                    |
| page_info    | <b>PageInfo</b> object              | 分页查询的信息                 |

表 4-136 Propagation

| 参数         | 参数类型   | 描述     |
|------------|--------|--------|
| id         | String | 关联唯一标识 |
| project_id | String | 项目ID   |

| 参数             | 参数类型                     | 描述                                                                                                                                                                                                     |
|----------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| er_id          | String                   | 企业路由器id                                                                                                                                                                                                |
| route_table_id | String                   | 路由表唯一标识                                                                                                                                                                                                |
| attachment_id  | String                   | 连接唯一标识                                                                                                                                                                                                 |
| resource_type  | String                   | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接，通过云连接CC加载不同区域的企业路由器来创建“对等连接（Peering）”连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| resource_id    | String                   | 连接的资源唯一标识                                                                                                                                                                                              |
| route_policy   | ImportRoutePolicy object | 入方向路由策略                                                                                                                                                                                                |
| state          | String                   | 状态:pending available modifying deleting deleted failed                                                                                                                                                 |
| created_at     | String                   | 资源创建时间<br>采用UTC时间<br>格式: YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                      |
| updated_at     | String                   | 资源更新时间<br>采用UTC时间<br>格式: YYYY-MM-DDTHH:mm:ss.sssZ                                                                                                                                                      |

表 4-137 ImportRoutePolicy

| 参数               | 参数类型   | 描述                                     |
|------------------|--------|----------------------------------------|
| import_policy_id | String | 入方向Ipv4协议路由策略id<br>最小长度: 0<br>最大长度: 36 |

表 4-138 PageInfo

| 参数          | 参数类型   | 描述                    |
|-------------|--------|-----------------------|
| next_marker | String | 下一个marker标识，为空时表示最后一页 |

| 参数            | 参数类型    | 描述        |
|---------------|---------|-----------|
| current_count | Integer | 当前列表中资源数量 |

## 请求示例

### 查询路由表传播列表

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/7dec531a-7714-11eb-9439-0242ac130002/route-tables/91c013e2-d65a-474e-9177-c3e1799ca726/propagations
```

## 响应示例

状态码：200

OK

```
{
  "propagations": [ {
    "id": "915a14a6-867b-4af7-83d1-70efceb146f9",
    "er_id": "7dec531a-7714-11eb-9439-0242ac130002",
    "route_table_id": "91c013e2-d65a-474e-9177-c3e1799ca726",
    "attachment_id": "a5b7d209-dc02-4c46-a51f-805eadd3de64",
    "resource_type": "vpc",
    "resource_id": "4e5fe97c-82bc-432e-87d8-06d7e157dffa",
    "route_policy": {
      "import_policy_id": ""
    },
    "state": "pending",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  } ],
  "page_info": {
    "next_marker": "1",
    "current_count": 1
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListPropagationsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
ListPropagationsRequest request = new ListPropagationsRequest();
request.withErId("{er_id}");
request.withRouteTableId("{route_table_id}");
try {
    ListPropagationsResponse response = client.listPropagations(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdr.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdr.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListPropagationsRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        response = client.list_propagations(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListPropagationsRequest{}
    request.ErId = "{er_id}"
    request.RouteTableId = "{route_table_id}"
    response, err := client.ListPropagations(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.6.3 删除路由传播

功能介绍

解绑连接和路由表的传播关系。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/route-tables/{route\_table\_id}/disable-propagations

表 4-139 路径参数

| 参数             | 是否必选 | 参数类型   | 描述        |
|----------------|------|--------|-----------|
| project_id     | 是    | String | 项目ID      |
| er_id          | 是    | String | 企业路由器实例ID |
| route_table_id | 是    | String | 路由表ID     |

请求参数

表 4-140 请求 Body 参数

| 参数            | 是否必选 | 参数类型                                      | 描述      |
|---------------|------|-------------------------------------------|---------|
| attachment_id | 否    | String                                    | 连接唯一标识  |
| route_policy  | 否    | <a href="#">ImportRoute Policy</a> object | 入方向路由策略 |

表 4-141 ImportRoutePolicy

| 参数               | 是否必选 | 参数类型   | 描述                                   |
|------------------|------|--------|--------------------------------------|
| import_policy_id | 否    | String | 入方向Ipv4协议路由策略id<br>最小长度：0<br>最大长度：36 |

## 响应参数

无

## 请求示例

删除路由表传播

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/ad838a5e-
dbef-22cc-b1d9-cb46bef77ae8/route-tables/915a14a6-867b-4af7-83d1-70efceb146f9/disable-propagations
{
  "attachment_id" : "a5b7d209-dc02-4c46-a51f-805eadd3de64"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除路由表传播

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DisablePropagationSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DisablePropagationRequest request = new DisablePropagationRequest();
        request.withErId("{er_id}");
        request.withRouteTableId("{route_table_id}");
        PropagationRequestBody body = new PropagationRequestBody();
```

```
body.withAttachmentId("a5b7d209-dc02-4c46-a51f-805eadd3de64");
request.withBody(body);
try {
    DisablePropagationResponse response = client.disablePropagation(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

### 删除路由表传播

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisablePropagationRequest()
        request.er_id = "{er_id}"
        request.route_table_id = "{route_table_id}"
        request.body = PropagationRequestBody(
            attachment_id="a5b7d209-dc02-4c46-a51f-805eadd3de64"
        )
        response = client.disable_propagation(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

### 删除路由表传播

```
package main
```

```
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    // variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
    projectId := "{project_id}"  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        WithProjectId(projectId).  
        Build()  
  
    client := er.NewErClient(  
        er.ErClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).  
            Build())  
  
    request := &model.DisablePropagationRequest{}  
    request.ErId = "{er_id}"  
    request.RouteTableId = "{route_table_id}"  
    attachmentIdPropagationRequestBody := "a5b7d209-dc02-4c46-a51f-805eadd3de64"  
    request.Body = &model.PropagationRequestBody{  
        AttachmentId: &attachmentIdPropagationRequestBody,  
    }  
    response, err := client.DisablePropagation(request)  
    if err == nil {  
        fmt.Printf("%v\n", response)  
    } else {  
        fmt.Println(err)  
    }  
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.7 路由

### 4.7.1 创建静态路由

#### 功能介绍

创建静态路由。静态路由为手动创建的路由，有效路由为优选后的路由。

#### 接口约束

当is\_blackhole设置为false时，必须传入attachment\_id参数;当is\_blackhole设置为true时，禁止传入attachment\_id参数。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/enterprise-router/route-tables/{route\_table\_id}/static-routes

表 4-142 路径参数

| 参数             | 是否必选 | 参数类型   | 描述    |
|----------------|------|--------|-------|
| project_id     | 是    | String | 项目ID  |
| route_table_id | 是    | String | 路由表ID |

#### 请求参数

表 4-143 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-144 请求 Body 参数

| 参数    | 是否必选 | 参数类型               | 描述      |
|-------|------|--------------------|---------|
| route | 是    | CreateRoute object | 创建路由请求体 |

表 4-145 CreateRoute

| 参数            | 是否必选 | 参数类型    | 描述                           |
|---------------|------|---------|------------------------------|
| destination   | 是    | String  | 路由目的地址<br>最小长度：0<br>最大长度：256 |
| attachment_id | 否    | String  | 路由下一跳指向的连接ID                 |
| is_blackhole  | 否    | Boolean | 是否为黑洞路由，默认为false             |
| description   | 否    | String  | 路由描述信息<br>最小长度：0<br>最大长度：255 |

响应参数

状态码：202

表 4-146 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-147 响应 Body 参数

| 参数         | 参数类型         | 描述   |
|------------|--------------|------|
| route      | Route object | 路由表项 |
| request_id | String       | 请求ID |

表 4-148 Route

| 参数           | 参数类型    | 描述                                                       |
|--------------|---------|----------------------------------------------------------|
| id           | String  | 路由id                                                     |
| type         | String  | 路由类型:static                                              |
| state        | String  | 路由状态:pending available modifying deleting deleted failed |
| is_blackhole | Boolean | 是否为黑洞路由                                                  |
| destination  | String  | 路由目的地址                                                   |

| 参数             | 参数类型                                             | 描述                               |
|----------------|--------------------------------------------------|----------------------------------|
| attachments    | Array of <a href="#">RouteAttachment</a> objects | 下一跳列表                            |
| route_table_id | String                                           | 路由表id                            |
| created_at     | String                                           | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| updated_at     | String                                           | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| description    | String                                           | 路由描述信息                           |

表 4-149 RouteAttachment

| 参数            | 参数类型   | 描述                                                                                                                                                                                                                     |
|---------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id   | String | 连接关联的资源ID                                                                                                                                                                                                              |
| resource_type | String | 连接关联的资源类型: <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>cfw: 云防火墙</li><li>ecn: 企业连接网</li></ul> |
| attachment_id | String | 连接ID                                                                                                                                                                                                                   |

请求示例

在企业路由器的路由表中创建静态路由, 该路由不是黑洞路由, 目的地址为192.168.0.0/16, 下一跳为指定连接

```
POST https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/route-tables/0d1748a0-5188-11e5-b86f-a51b5f125b84/static-routes
{
  "route": {
    "destination": "192.168.0.0/16",
    "is_blackhole": false,
    "attachment_id": "b70aee08-c671-4cad-9fd5-7381d163bcc8"
  }
}
```

## 响应示例

状态码：202

Accepted

```
{
  "route": {
    "id": "9b3b38a9-1c9d-4f01-9429-81af6b545289",
    "type": "static",
    "destination": "192.168.0.0/16",
    "description": "static route",
    "attachments": [ {
      "resource_id": "9c4ab06a-0ab4-4fad-93a8-f733f3a4433d",
      "resource_type": "vpc",
      "attachment_id": "b70aee08-c671-4cad-9fd5-7381d163bcc8"
    } ],
    "route_table_id": "0d1748a0-5188-11e5-b86f-a51b5f125b84",
    "is_blackhole": false,
    "state": "pending",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

在企业路由器的路由表中创建静态路由，该路由不是黑洞路由，目的地址为192.168.0.0/16，下一跳为指定连接

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class CreateStaticRouteSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
```

```
        .withRegion(ErRegion.valueOf("<YOUR REGION>"))
        .build();
CreateStaticRouteRequest request = new CreateStaticRouteRequest();
request.withRouteTableId("{route_table_id}");
CreateRouteRequestBody body = new CreateRouteRequestBody();
CreateRoute routebody = new CreateRoute();
routebody.withDestination("192.168.0.0/16")
        .withAttachmentId("b70aee08-c671-4cad-9fd5-7381d163bcc8")
        .withIsBlackhole(false);
body.withRoute(routebody);
request.withBody(body);
try {
    CreateStaticRouteResponse response = client.createStaticRoute(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

在企业路由器的路由表中创建静态路由，该路由不是黑洞路由，目的地址为 192.168.0.0/16，下一跳为指定连接

# coding: utf-8

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.valueOf("<YOUR REGION>")) \
        .build()

    try:
        request = CreateStaticRouteRequest()
        request.route_table_id = "{route_table_id}"
        routebody = CreateRoute(
            destination="192.168.0.0/16",
            attachment_id="b70aee08-c671-4cad-9fd5-7381d163bcc8",
            is_blackhole=False
        )
        request.body = CreateRouteRequestBody(
            route=routebody
        )
```

```
response = client.create_static_route(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

在企业路由器的路由表中创建静态路由，该路由不是黑洞路由，目的地址为 192.168.0.0/16，下一跳为指定连接

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateStaticRouteRequest{}
    request.RouteTableId = "{route_table_id}"
    attachmentIdRoute := "b70aee08-c671-4cad-9fd5-7381d163bcc8"
    isBlackholeRoute := false
    routebody := &model.CreateRoute{
        Destination: "192.168.0.0/16",
        AttachmentId: &attachmentIdRoute,
        IsBlackhole: &isBlackholeRoute,
    }
    request.Body = &model.CreateRouteRequestBody{
        Route: routebody,
    }
    response, err := client.CreateStaticRoute(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.7.2 更新静态路由

### 功能介绍

更新静态路由

### 接口约束

当is\_blackhole设置为false时，必须传入attachment\_id参数;当is\_blackhole设置为true时，禁止传入attachment\_id参数。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v3/{project\_id}/enterprise-router/route-tables/{route\_table\_id}/static-routes/{route\_id}

表 4-150 路径参数

| 参数             | 是否必选 | 参数类型   | 描述    |
|----------------|------|--------|-------|
| project_id     | 是    | String | 项目ID  |
| route_table_id | 是    | String | 路由表ID |
| route_id       | 是    | String | 路由ID  |

## 请求参数

表 4-151 请求 Body 参数

| 参数    | 是否必选 | 参数类型                               | 描述 |
|-------|------|------------------------------------|----|
| route | 是    | <a href="#">UpdateRoute</a> object | 路由 |

表 4-152 UpdateRoute

| 参数            | 是否必选 | 参数类型    | 描述                           |
|---------------|------|---------|------------------------------|
| attachment_id | 否    | String  | 路由下一跳                        |
| is_blackhole  | 否    | Boolean | 是否为黑洞路由                      |
| description   | 否    | String  | 路由描述信息<br>最小长度：0<br>最大长度：255 |

## 响应参数

状态码：202

表 4-153 响应 Body 参数

| 参数         | 参数类型                         | 描述   |
|------------|------------------------------|------|
| route      | <a href="#">Route</a> object | 路由表项 |
| request_id | String                       | 请求ID |

表 4-154 Route

| 参数           | 参数类型    | 描述                                                       |
|--------------|---------|----------------------------------------------------------|
| id           | String  | 路由id                                                     |
| type         | String  | 路由类型:static                                              |
| state        | String  | 路由状态:pending available modifying deleting deleted failed |
| is_blackhole | Boolean | 是否为黑洞路由                                                  |
| destination  | String  | 路由目的地址                                                   |

| 参数             | 参数类型                                             | 描述                               |
|----------------|--------------------------------------------------|----------------------------------|
| attachments    | Array of <a href="#">RouteAttachment</a> objects | 下一跳列表                            |
| route_table_id | String                                           | 路由表id                            |
| created_at     | String                                           | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| updated_at     | String                                           | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| description    | String                                           | 路由描述信息                           |

表 4-155 RouteAttachment

| 参数            | 参数类型   | 描述                                                                                                                                                                                                                                 |
|---------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id   | String | 连接关联的资源ID                                                                                                                                                                                                                          |
| resource_type | String | 连接关联的资源类型: <ul style="list-style-type: none"><li>• vpc: 虚拟私有云</li><li>• vpn: vpn网关</li><li>• vgw: 云专线的虚拟网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>• cfw: 云防火墙</li><li>• ecn: 企业连接网</li></ul> |
| attachment_id | String | 连接ID                                                                                                                                                                                                                               |

请求示例

在企业路由器的路由表中修改静态路由，将黑洞路由修改为常规路由，下一跳为指定连接

```
PUT https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbef3c/enterprise-router/route-tables/19d334b7-78c1-4e0e-ba29-b797e641e23c/static-routes/9b3b38a9-1c9d-4f01-9429-81af6b545289

{
  "route": {
    "is_blackhole": false,
    "attachment_id": "b70aee08-c671-4cad-9fd5-7381d163bcc8"
  }
}
```

响应示例

状态码：202

## Accepted

```
{
  "route": {
    "id": "9b3b38a9-1c9d-4f01-9429-81af6b545289",
    "type": "static",
    "destination": "192.168.0.0/16",
    "attachments": [ {
      "resource_id": "9c4ab06a-0ab4-4fad-93a8-f733f3a4433d",
      "resource_type": "vpc",
      "attachment_id": "b70aee08-c671-4cad-9fd5-7381d163bcc8"
    } ],
    "route_table_id": "19d334b7-78c1-4e0e-ba29-b797e641e23c",
    "is_blackhole": false,
    "state": "pending",
    "created_at": "2020-03-11T15:13:31",
    "updated_at": "2020-03-11T15:13:31"
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

在企业路由器的路由表中修改静态路由，将黑洞路由修改为常规路由，下一跳为指定连接

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class UpdateStaticRouteSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        UpdateStaticRouteRequest request = new UpdateStaticRouteRequest();
        request.withRouteTableId("{route_table_id}");
        request.withRouteId("{route_id}");
        UpdateRouteRequestBody body = new UpdateRouteRequestBody();
```

```
UpdateRoute routebody = new UpdateRoute();
routebody.withAttachmentId("b70aee08-c671-4cad-9fd5-7381d163bcc8")
    .withIsBlackhole(false);
body.withRoute(routebody);
request.withBody(body);
try {
    UpdateStaticRouteResponse response = client.updateStaticRoute(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

在企业路由器的路由表中修改静态路由，将黑洞路由修改为常规路由，下一跳为指定连接

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateStaticRouteRequest()
        request.route_table_id = "{route_table_id}"
        request.route_id = "{route_id}"
        routebody = UpdateRoute(
            attachment_id="b70aee08-c671-4cad-9fd5-7381d163bcc8",
            is_blackhole=False
        )
        request.body = UpdateRouteRequestBody(
            route=routebody
        )
        response = client.update_static_route(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

在企业路由器的路由表中修改静态路由，将黑洞路由修改为常规路由，下一跳为指定连接

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.NewErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateStaticRouteRequest{}
    request.RouteTableId = "{route_table_id}"
    request.RouteId = "{route_id}"
    attachmentIdRoute := "b70aee08-c671-4cad-9fd5-7381d163bcc8"
    isBlackholeRoute := false
    routebody := &model.UpdateRoute{
        AttachmentId: &attachmentIdRoute,
        IsBlackhole: &isBlackholeRoute,
    }
    request.Body = &model.UpdateRouteRequestBody{
        Route: routebody,
    }
    response, err := client.UpdateStaticRoute(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.7.3 查询静态路由详情

### 功能介绍

查询静态路由详情

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/route-tables/{route\_table\_id}/static-routes/{route\_id}

表 4-156 路径参数

| 参数             | 是否必选 | 参数类型   | 描述    |
|----------------|------|--------|-------|
| project_id     | 是    | String | 项目ID  |
| route_table_id | 是    | String | 路由表ID |
| route_id       | 是    | String | 路由ID  |

### 请求参数

无

### 响应参数

状态码：200

表 4-157 响应 Body 参数

| 参数         | 参数类型                         | 描述   |
|------------|------------------------------|------|
| route      | <a href="#">Route</a> object | 路由表项 |
| request_id | String                       | 请求ID |

表 4-158 Route

| 参数             | 参数类型                                             | 描述                                                       |
|----------------|--------------------------------------------------|----------------------------------------------------------|
| id             | String                                           | 路由id                                                     |
| type           | String                                           | 路由类型:static                                              |
| state          | String                                           | 路由状态:pending available modifying deleting deleted failed |
| is_blackhole   | Boolean                                          | 是否为黑洞路由                                                  |
| destination    | String                                           | 路由目的地址                                                   |
| attachments    | Array of <a href="#">RouteAttachment</a> objects | 下一跳列表                                                    |
| route_table_id | String                                           | 路由表id                                                    |
| created_at     | String                                           | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| updated_at     | String                                           | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                         |
| description    | String                                           | 路由描述信息                                                   |

表 4-159 RouteAttachment

| 参数            | 参数类型   | 描述                                                                                                                                                                                                                                 |
|---------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id   | String | 连接关联的资源ID                                                                                                                                                                                                                          |
| resource_type | String | 连接关联的资源类型: <ul style="list-style-type: none"><li>• vpc: 虚拟私有云</li><li>• vpn: vpn网关</li><li>• vgw: 云专线的虚拟网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>• cfw: 云防火墙</li><li>• ecn: 企业连接网</li></ul> |
| attachment_id | String | 连接ID                                                                                                                                                                                                                               |

## 请求示例

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/route-tables/19d334b7-78c1-4e0e-ba29-b797e641e23c/static-routes/19d334b7-78c1-4e0e-ba29-b797e641e23c
```

## 响应示例

状态码：200

OK

```
{
  "route": {
    "id": "19d334b7-78c1-4e0e-ba29-b797e641e23c",
    "type": "static",
    "destination": "192.168.0.0/16",
    "description": "static route",
    "attachments": [ {
      "resource_id": "9c4ab06a-0ab4-4fad-93a8-f733f3a4433d",
      "resource_type": "VPC",
      "attachment_id": "9b3b38a9-1c9d-4f01-9429-81af6b545289"
    } ],
    "route_table_id": "19d334b7-78c1-4e0e-ba29-b797e641e23c",
    "is_blackhole": false,
    "state": "pending",
    "created_at": "2020-03-11T15:13:31",
    "updated_at": "2020-03-11T15:13:31"
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ShowStaticRouteSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
```

```
        .withCredential(auth)
        .withRegion(ErRegion.valueOf("<YOUR REGION>"))
        .build();
ShowStaticRouteRequest request = new ShowStaticRouteRequest();
request.withRouteTableId("{route_table_id}");
request.withRouteId("{route_id}");
try {
    ShowStaticRouteResponse response = client.showStaticRoute(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowStaticRouteRequest()
        request.route_table_id = "{route_table_id}"
        request.route_id = "{route_id}"
        response = client.show_static_route(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
```

```
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowStaticRouteRequest{}
    request.RouteTableId = "{route_table_id}"
    request.RouteId = "{route_id}"
    response, err := client.ShowStaticRoute(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.7.4 查询静态路由列表

功能介绍

查询静态路由列表。

接口约束

- 支持分页查询，支持过滤查询：destination，attachment\_id, resource\_type, type。
- 支持单字段排序，排序字段有[id,destination,created\_at,updated\_at]，不支持多字段排序。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project\_id}/enterprise-router/route-tables/{route\_table\_id}/static-routes

表 4-160 路径参数

| 参数             | 是否必选 | 参数类型   | 描述    |
|----------------|------|--------|-------|
| project_id     | 是    | String | 项目ID  |
| route_table_id | 是    | String | 路由表ID |

表 4-161 Query 参数

| 参数            | 是否必选 | 参数类型             | 描述                                                                             |
|---------------|------|------------------|--------------------------------------------------------------------------------|
| limit         | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                  |
| marker        | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b> |
| destination   | 否    | Array of strings | 路由目的地址                                                                         |
| attachment_id | 否    | Array of strings | 连接ID                                                                           |

| 参数            | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                       |
|---------------|------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | 否    | Array of strings | <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 (Peering)”连接</li><li>ecn: 企业连接网</li><li>cfw: 云防火墙</li></ul> |
| sort_key      | 否    | Array of strings | 按关键字排序, 默认按照id排序, 可选值:id name state                                                                                                                                                                      |
| sort_dir      | 否    | Array of strings | 返回结果按照升序或降序排列, 默认为asc,降序为desc                                                                                                                                                                            |

请求参数

无

响应参数

状态码：200

表 4-162 响应 Body 参数

| 参数         | 参数类型                          | 描述      |
|------------|-------------------------------|---------|
| routes     | Array of <b>Route</b> objects | 路由列表    |
| request_id | String                        | 请求ID    |
| page_info  | <b>PageInfo</b> object        | 分页查询的信息 |

表 4-163 Route

| 参数           | 参数类型    | 描述                                                       |
|--------------|---------|----------------------------------------------------------|
| id           | String  | 路由id                                                     |
| type         | String  | 路由类型:static                                              |
| state        | String  | 路由状态:pending available modifying deleting deleted failed |
| is_blackhole | Boolean | 是否为黑洞路由                                                  |

| 参数             | 参数类型                                             | 描述                               |
|----------------|--------------------------------------------------|----------------------------------|
| destination    | String                                           | 路由目的地址                           |
| attachments    | Array of <a href="#">RouteAttachment</a> objects | 下一跳列表                            |
| route_table_id | String                                           | 路由表id                            |
| created_at     | String                                           | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| updated_at     | String                                           | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ |
| description    | String                                           | 路由描述信息                           |

表 4-164 RouteAttachment

| 参数            | 参数类型   | 描述                                                                                                                                                                                                                     |
|---------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id   | String | 连接关联的资源ID                                                                                                                                                                                                              |
| resource_type | String | 连接关联的资源类型: <ul style="list-style-type: none"><li>vpc: 虚拟私有云</li><li>vpn: vpn网关</li><li>vgw: 云专线的虚拟网关</li><li>peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>cfw: 云防火墙</li><li>ecn: 企业连接网</li></ul> |
| attachment_id | String | 连接ID                                                                                                                                                                                                                   |

表 4-165 PageInfo

| 参数            | 参数类型    | 描述                     |
|---------------|---------|------------------------|
| next_marker   | String  | 下一个marker标识, 为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量              |

请求示例

查询静态路由列表

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/route-tables/915a14a6-867b-4af7-83d1-70efceb146f5/static-routes
```

## 响应示例

状态码：200

OK

```
{
  "routes": [ {
    "id": "9b3b38a9-1c9d-4f01-9429-81af6b545289",
    "type": "static",
    "destination": "192.168.0.0/16",
    "description": "static route",
    "attachments": [ {
      "resource_id": "9c4ab06a-0ab4-4fad-93a8-f733f3a4433d",
      "resource_type": "vpc",
      "attachment_id": "9b3b38a9-1c9d-4f01-9429-81af6b545289"
    } ],
    "route_table_id": "19d334b7-78c1-4e0e-ba29-b797e641e23c",
    "state": "pending",
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2020-03-11T15:13:31Z"
  } ],
  "page_info": {
    "next_marker": "1",
    "current_count": 1
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListStaticRoutesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);
```

```
ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
ListStaticRoutesRequest request = new ListStaticRoutesRequest();
request.withRouteTableId("{route_table_id}");
try {
    ListStaticRoutesResponse response = client.listStaticRoutes(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListStaticRoutesRequest()
        request.route_table_id = "{route_table_id}"
        response = client.list_static_routes(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListStaticRoutesRequest{}
    request.RouteTableId = "{route_table_id}"
    response, err := client.ListStaticRoutes(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.7.5 查询有效路由列表

功能介绍

查询有效的路由列表，支持分页查询能力。有效路由为优选后的路由。

调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{project\_id}/enterprise-router/route-tables/{route\_table\_id}/routes

表 4-166 路径参数

| 参数             | 是否必选 | 参数类型   | 描述    |
|----------------|------|--------|-------|
| project_id     | 是    | String | 项目ID  |
| route_table_id | 是    | String | 路由表ID |

表 4-167 Query 参数

| 参数            | 是否必选 | 参数类型             | 描述                                                                                                                                                                                               |
|---------------|------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| limit         | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                                                                                                                                    |
| marker        | 否    | String           | 上一页最后一条记录的路由id，<br>为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b>                                                                                                        |
| destination   | 否    | Array of strings | 路由目的地址                                                                                                                                                                                           |
| resource_type | 否    | Array of strings | <ul style="list-style-type: none"><li>vpc：虚拟私有云</li><li>vpn：vpn网关</li><li>vgw：云专线的虚拟网关</li><li>peering：对等连接，通过云连接CC加载不同区域的企业路由器来创建“对等连接（Peering）”连接</li><li>ecn：企业连接网</li><li>cfw：云防火墙</li></ul> |

## 请求参数

无

## 响应参数

状态码：200

表 4-168 响应 Body 参数

| 参数         | 参数类型                                            | 描述      |
|------------|-------------------------------------------------|---------|
| routes     | Array of <a href="#">EffectiveRoute</a> objects | 路由列表    |
| request_id | String                                          | 请求ID    |
| page_info  | <a href="#">PageInfo</a> object                 | 分页查询的信息 |

表 4-169 EffectiveRoute

| 参数           | 参数类型                                                      | 描述                      |
|--------------|-----------------------------------------------------------|-------------------------|
| route_id     | String                                                    | 路由ID                    |
| destination  | String                                                    | 路由目的地                   |
| next_hops    | Array of <a href="#">EffectiveRouteAttachment</a> objects | 路由下一跳列表                 |
| is_blackhole | Boolean                                                   | 是否黑洞路由                  |
| route_type   | String                                                    | 路由类型:static,propagation |
| description  | String                                                    | 路由描述信息                  |

表 4-170 EffectiveRouteAttachment

| 参数            | 参数类型   | 描述                                                                                                                                                                                                                                 |
|---------------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id   | String | 连接关联的资源ID                                                                                                                                                                                                                          |
| resource_type | String | 连接关联的资源类型: <ul style="list-style-type: none"><li>• vpc: 虚拟私有云</li><li>• vpn: vpn网关</li><li>• vgw: 云专线的虚拟网关</li><li>• peering: 对等连接, 通过云连接CC加载不同区域的企业路由器来创建“对等连接 ( Peering )” 连接</li><li>• cfw: 云防火墙</li><li>• ecn: 企业连接网</li></ul> |
| attachment_id | String | 连接ID                                                                                                                                                                                                                               |

表 4-171 PageInfo

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| next_marker   | String  | 下一个marker标识，为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量             |

### 请求示例

查询有效路由列表

```
GET https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/route-tables/915a14a6-867b-4af7-83d1-70efceb146f9/routes
```

### 响应示例

状态码：200

OK

```
{
  "routes": [ {
    "route_id": "9b3b38a9-1c9d-4f01-9429-81af6b545289",
    "destination": "192.168.0.0/16",
    "description": "static route",
    "next_hops": [ {
      "resource_id": "9c4ab06a-0ab4-4fad-93a8-f733f3a4433d",
      "resource_type": "vpc",
      "attachment_id": "9b3b38a9-1c9d-4f01-9429-81af6b545289"
    } ],
    "is_blackhole": false,
    "route_type": "static"
  } ],
  "page_info": {
    "next_marker": "1",
    "current_count": 1
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

### SDK 代码示例

SDK代码示例如下。

#### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListEffectiveRoutesSolution {
```

```
public static void main(String[] args) {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running
    // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    String ak = System.getenv("CLOUD_SDK_AK");
    String sk = System.getenv("CLOUD_SDK_SK");
    String projectId = "{project_id}";

    ICredential auth = new BasicCredentials()
        .withProjectId(projectId)
        .withAk(ak)
        .withSk(sk);

    ErClient client = ErClient.newBuilder()
        .withCredential(auth)
        .withRegion(ErRegion.valueOf("<YOUR REGION>"))
        .build();
    ListEffectiveRoutesRequest request = new ListEffectiveRoutesRequest();
    request.withRouteTableId("{route_table_id}");
    try {
        ListEffectiveRoutesResponse response = client.listEffectiveRoutes(request);
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListEffectiveRoutesRequest()
        request.route_table_id = "{route_table_id}"
        response = client.list_effective_routes(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListEffectiveRoutesRequest{}
    request.RouteTableId = "{route_table_id}"
    response, err := client.ListEffectiveRoutes(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.7.6 删除静态路由

### 功能介绍

删除静态路由

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v3/{project\_id}/enterprise-router/route-tables/{route\_table\_id}/static-routes/{route\_id}

表 4-172 路径参数

| 参数             | 是否必选 | 参数类型   | 描述    |
|----------------|------|--------|-------|
| project_id     | 是    | String | 项目ID  |
| route_table_id | 是    | String | 路由表ID |
| route_id       | 是    | String | 路由ID  |

### 请求参数

无

### 响应参数

无

### 请求示例

```
DELETE https://{erouter_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/route-tables/3ddb4dc2-174e-4d43-9328-b805d5c0c774/static-routes/19d334b7-78c1-4e0e-ba29-b797e641e23c
```

### 响应示例

无

### SDK 代码示例

SDK代码示例如下。

#### Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DeleteStaticRouteSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteStaticRouteRequest request = new DeleteStaticRouteRequest();
        request.withRouteTableId("{route_table_id}");
        request.withRouteId("{route_id}");
        try {
            DeleteStaticRouteResponse response = client.deleteStaticRoute(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
```

```
projectId = "{project_id}"

credentials = BasicCredentials(ak, sk, projectId)

client = ErClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(ErRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteStaticRouteRequest()
    request.route_table_id = "{route_table_id}"
    request.route_id = "{route_id}"
    response = client.delete_static_route(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteStaticRouteRequest{}
    request.RouteTableId = "{route_table_id}"
    request.RouteId = "{route_id}"
    response, err := client.DeleteStaticRoute(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

# 4.8 标签

## 4.8.1 查询项目标签

### 功能介绍

查询特定类型资源的标签集合。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/{resource\_type}/tags

表 4-173 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                         |
|---------------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | 是    | String | 标签资源类型: <ul style="list-style-type: none"><li>instance: 企业路由器实例</li><li>route-table: 路由表</li><li>vpc-attachment: VPC连接</li><li>vgw-attachment: 虚拟网关连接</li><li>peering-attachment: 对等连接（Peering）连接</li><li>vpn-attachment: VPN网关连接</li><li>ecn-attachment: 企业连接网连接</li><li>cfw-attachment: 云防火墙连接</li></ul> |

请求参数

无

响应参数

状态码：200

表 4-174 响应 Body 参数

| 参数   | 参数类型                         | 描述    |
|------|------------------------------|-------|
| tags | Array of <b>Tags</b> objects | 标签列表。 |

表 4-175 Tags

| 参数     | 参数类型             | 描述                          |
|--------|------------------|-----------------------------|
| key    | String           | 标签键，不能为空，最大长度127个unicode字符。 |
| values | Array of strings | 值列表，每个值最大长度255个unicode字符。   |

请求示例

查询企业路由器标签

https://{erouter\_endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/tags

## 响应示例

状态码：200

OK

```
{
  "tags": [ {
    "key": "keys",
    "values": [ "value" ]
  }, {
    "key": "key3",
    "values": [ "value3", "value33" ]
  }, {
    "key": "key1",
    "values": [ "value1" ]
  }, {
    "key": "key2",
    "values": [ "value2", "value22" ]
  } ]
}
```

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.8.2 查询资源标签

### 功能介绍

查询特定类型资源的标签信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/{resource\_type}/{resource\_id}/tags

表 4-176 路径参数

| 参数          | 是否必选 | 参数类型   | 描述   |
|-------------|------|--------|------|
| project_id  | 是    | String | 项目ID |
| resource_id | 是    | String | 资源ID |

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                         |
|---------------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_type | 是    | String | 标签资源类型: <ul style="list-style-type: none"><li>• instance: 企业路由器实例</li><li>• route-table: 路由表</li><li>• vpc-attachment: VPC连接</li><li>• vgw-attachment: 虚拟网关连接</li><li>• peering-attachment: 对等连接（Peering）连接</li><li>• vpn-attachment: VPN网关连接</li><li>• ecn-attachment: 企业连接网连接</li><li>• cfw-attachment: 云防火墙连接</li></ul> |

请求参数

无

响应参数

状态码：200

表 4-177 响应 Body 参数

| 参数   | 参数类型                        | 描述   |
|------|-----------------------------|------|
| tags | Array of <b>Tag</b> objects | 标签列表 |

表 4-178 Tag

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                          |
|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母(包含中文)、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                          |

请求示例

查询企业路由器实例标签

```
https://{erouter_endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/f1a28dfd-186f-4625-b6b1-f05e5e8609c0/tags
```

响应示例

状态码：200

OK

```
{
  "tags": [ {
    "key": "key2",
    "value": "value2"
  }, {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key3",
    "value": "value3"
  } ]
}
```

```
    }  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.er.v3.region.ErRegion;  
import com.huaweicloud.sdk.er.v3.*;  
import com.huaweicloud.sdk.er.v3.model.*;  
  
public class ShowResourceTagSolution {  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        ErClient client = ErClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))  
            .build();  
        ShowResourceTagRequest request = new ShowResourceTagRequest();  
        request.withResourceId("{resource_id}");  
  
        request.withResourceType(ShowResourceTagRequest.ResourceTypeEnum.fromValue("{resource_type}"));  
        try {  
            ShowResourceTagResponse response = client.showResourceTag(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrMsg());  
        }  
    }  
}
```

### Python

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowResourceTagRequest()
        request.resource_id = "{resource_id}"
        request.resource_type = "{resource_type}"
        response = client.show_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```
request := &model.ShowResourceTagRequest{}
request.ResourceId = "{resource_id}"
request.ResourceType = model.GetShowResourceTagRequestResourceTypeEnum().RESOURCE_TYPE
response, err := client.ShowResourceTag(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.8.3 创建资源标签

功能介绍

为特定类型的资源创建标签。

接口约束

- 一个资源上最多有20个标签。
- 此接口为幂等接口。
- 创建时，如果创建的标签已经存在（key相同），则覆盖。

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project\_id}/{resource\_type}/{resource\_id}/tags

表 4-179 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                         |
|---------------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| resource_id   | 是    | String | 资源ID                                                                                                                                                                                                                                                                                                                       |
| resource_type | 是    | String | 标签资源类型: <ul style="list-style-type: none"><li>• instance: 企业路由器实例</li><li>• route-table: 路由表</li><li>• vpc-attachment: VPC连接</li><li>• vgw-attachment: 虚拟网关连接</li><li>• peering-attachment: 对等连接（Peering）连接</li><li>• vpn-attachment: VPN网关连接</li><li>• ecn-attachment: 企业连接网连接</li><li>• cfw-attachment: 云防火墙连接</li></ul> |

请求参数

表 4-180 请求 Body 参数

| 参数  | 是否必选 | 参数类型       | 描述   |
|-----|------|------------|------|
| tag | 是    | Tag object | 资源标签 |

表 4-181 Tag

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                 |
|-------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | 否    | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线 “_”，点 “.”，冒号 “:”，等号 “=”，加号 “+”，中划线 “-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | 否    | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线 “_”，点 “.”，冒号 “:”，等号 “=”，加号 “+”，中划线 “-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

响应参数

无

请求示例

为企业路由器资源添加标签，标签值为key1，标签键为value1

```
POST https://{erouter-endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags

{
  "tag": {
    "key": "key1",
    "value": "value1"
  }
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

为企业路由器资源添加标签，标签值为key1，标签键为value1

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class CreateResourceTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateResourceTagRequest request = new CreateResourceTagRequest();
        request.withResourceId("{resource_id}");

        request.withResourceType(CreateResourceTagRequest.ResourceTypeEnum.fromValue("{resource_type}"));
        CreateResourceTagRequestBody body = new CreateResourceTagRequestBody();
        Tag tagbody = new Tag();
        tagbody.withKey("key1")
            .withValue("value1");
        body.withTag(tagbody);
        request.withBody(body);
        try {
            CreateResourceTagResponse response = client.createResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
        }
    }
}
```

```
        System.out.println(e.getErrorMsg());
    }
}
```

## Python

为企业路由器资源添加标签，标签值为key1，标签键为value1

```
# coding: utf-8
```

```
import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateResourceTagRequest()
        request.resource_id = "{resource_id}"
        request.resource_type = "{resource_type}"
        tagbody = Tag(
            key="key1",
            value="value1"
        )
        request.body = CreateResourceTagRequestBody(
            tag=tagbody
        )
        response = client.create_resource_tag(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

为企业路由器资源添加标签，标签值为key1，标签键为value1

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")
projectId := "{project_id}"

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    WithProjectId(projectId).
    Build()

client := er.NewErClient(
    er.ClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.CreateResourceTagRequest{}
request.ResourceId = "{resource_id}"
request.ResourceType = model.GetCreateResourceTagRequestResourceTypeEnum().RESOURCE_TYPE
keyTag := "key1"
valueTag := "value1"
tagbody := &model.Tag{
    Key: &keyTag,
    Value: &valueTag,
}
request.Body = &model.CreateResourceTagRequestBody{
    Tag: tagbody,
}
response, err := client.CreateResourceTag(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述         |
|-----|------------|
| 204 | No Content |

错误码

请参见[错误码](#)。

## 4.8.4 批量添加删除资源标签

### 功能介绍

- 为指定实例批量添加或删除标签
- 标签管理服务需要使用该接口批量管理实例的标签。
- 一个资源上最多有20个标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/{resource\_type}/{resource\_id}/tags/action

表 4-182 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                         |
|---------------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| project_id    | 是    | String | 项目ID                                                                                                                                                                                                                                                                                                                       |
| resource_id   | 是    | String | 资源ID                                                                                                                                                                                                                                                                                                                       |
| resource_type | 是    | String | 标签资源类型: <ul style="list-style-type: none"><li>• instance: 企业路由器实例</li><li>• route-table: 路由表</li><li>• vpc-attachment: VPC连接</li><li>• vgw-attachment: 虚拟网关连接</li><li>• peering-attachment: 对等连接（Peering）连接</li><li>• vpn-attachment: VPN网关连接</li><li>• ecn-attachment: 企业连接网连接</li><li>• cfw-attachment: 云防火墙连接</li></ul> |

## 请求参数

表 4-183 请求 Body 参数

| 参数     | 是否必选 | 参数类型                        | 描述                                              |
|--------|------|-----------------------------|-------------------------------------------------|
| action | 是    | String                      | 功能说明：操作标识。<br>取值范围：<br>create（创建）<br>delete（删除） |
| tags   | 否    | Array of <b>Tag</b> objects | 标签列表。                                           |

表 4-184 Tag

| 参数    | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                           |
|-------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| key   | 否    | String | 标签键，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• _sys_ 开头属于系统标签，租户不能输入。</li><li>• 标签key不能为空（长度不能为0）。</li><li>• 长度在1-128之间。</li></ul> 最小长度：1<br>最大长度：128 |
| value | 否    | String | 标签值，规则： <ul style="list-style-type: none"><li>• 可用 UTF-8 格式表示的字母（包含中文）、数字和空格，以及字符：下划线“_”，点“.”，冒号“:”，等号“=”，加号“+”，中划线“-”，“@”。</li><li>• 资源标签值可以为空（empty or null）。</li><li>• 长度在0-255之间。</li></ul> 最小长度：0<br>最大长度：255                           |

## 响应参数

无

## 请求示例

- 批量添加资源标签。

POST https://{erouter-endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/action

```
{
  "action": "create",
  "tags": [ {
    "key": "key1",
    "value": "value1"
  }, {
    "key": "key2",
    "value": "value2"
  } ]
}
```

- 批量删除资源标签。

POST https://{erouter-endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/action

```
{
  "action": "delete",
  "tags": [ {
    "key": "key1"
  }, {
    "key": "key2",
    "value": "value3"
  } ]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

- 批量添加资源标签。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErrRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchCreateResourceTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
BatchCreateResourceTagsRequest request = new BatchCreateResourceTagsRequest();
request.withResourceId("{resource_id}");

request.withResourceType(BatchCreateResourceTagsRequest.ResourceTypeEnum.fromValue("{resource_type}"));
BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
List<Tag> listbodyTags = new ArrayList<>();
listbodyTags.add(
    new Tag()
        .withKey("key1")
        .withValue("value1")
);
listbodyTags.add(
    new Tag()
        .withKey("key2")
        .withValue("value2")
);
body.withTags(listbodyTags);
body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("create"));
request.withBody(body);
try {
    BatchCreateResourceTagsResponse response = client.batchCreateResourceTags(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 批量删除资源标签。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

import java.util.List;
import java.util.ArrayList;
```

```
public class BatchCreateResourceTagsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        BatchCreateResourceTagsRequest request = new BatchCreateResourceTagsRequest();
        request.withResourceId("{resource_id}");

        request.withResourceType(BatchCreateResourceTagsRequest.ResourceTypeEnum.fromValue("{resource_
        type}"));
        BatchOperateResourceTagsRequestBody body = new BatchOperateResourceTagsRequestBody();
        List<Tag> listbodyTags = new ArrayList<>();
        listbodyTags.add(
            new Tag()
                .withKey("key1")
        );
        listbodyTags.add(
            new Tag()
                .withKey("key2")
                .withValue("value3")
        );
        body.withTags(listbodyTags);
        body.withAction(BatchOperateResourceTagsRequestBody.ActionEnum.fromValue("delete"));
        request.withBody(body);
        try {
            BatchCreateResourceTagsResponse response = client.batchCreateResourceTags(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

- 批量添加资源标签。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *
```

```
if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = BatchCreateResourceTagsRequest()
        request.resource_id = "{resource_id}"
        request.resource_type = "{resource_type}"
        listTagsbody = [
            Tag(
                key="key1",
                value="value1"
            ),
            Tag(
                key="key2",
                value="value2"
            )
        ]
        request.body = BatchOperateResourceTagsRequestBody(
            tags=listTagsbody,
            action="create"
        )
        response = client.batch_create_resource_tags(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

- 批量删除资源标签。

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
    request = BatchCreateResourceTagsRequest()
    request.resource_id = "{resource_id}"
    request.resource_type = "{resource_type}"
    listTagsbody = [
        Tag(
            key="key1"
        ),
        Tag(
            key="key2",
            value="value3"
        )
    ]
    request.body = BatchOperateResourceTagsRequestBody(
        tags=listTagsbody,
        action="delete"
    )
    response = client.batch_create_resource_tags(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

- 批量添加资源标签。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateResourceTagsRequest{}
    request.ResourceId = "{resource_id}"
    request.ResourceType =
        model.GetBatchCreateResourceTagsRequestResourceTypeEnum().RESOURCE_TYPE
    keyTags:= "key1"
    valueTags:= "value1"
    keyTags1:= "key2"
```

```
valueTags1:= "value2"
var listTagsbody = []model.Tag{
    {
        Key: &keyTags,
        Value: &valueTags,
    },
    {
        Key: &keyTags1,
        Value: &valueTags1,
    },
}
request.Body = &model.BatchOperateResourceTagsRequestBody{
    Tags: &listTagsbody,
    Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().CREATE,
}
response, err := client.BatchCreateResourceTags(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 批量删除资源标签。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.BatchCreateResourceTagsRequest{}
    request.ResourceId = "{resource_id}"
    request.ResourceType =
        model.GetBatchCreateResourceTagsRequestResourceTypeEnum().RESOURCE_TYPE
    keyTags:= "key1"
    keyTags1:= "key2"
    valueTags:= "value3"
    var listTagsbody = []model.Tag{
        {
            Key: &keyTags,
        },
        {
            Key: &keyTags1,
```

```
        Value: &valueTags,
    },
}
request.Body = &model.BatchOperateResourceTagsRequestBody{
    Tags: &listTagsbody,
    Action: model.GetBatchOperateResourceTagsRequestBodyActionEnum().DELETE,
}
response, err := client.BatchCreateResourceTags(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述         |
|-----|------------|
| 204 | No Content |

错误码

请参见[错误码](#)。

4.8.5 删除资源标签

功能介绍

删除特定类型资源的标签。

接口约束

此接口为幂等接口，Key不能为空或者空字符串。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v3/{project\_id}/{resource\_type}/{resource\_id}/tags/{key}

表 4-185 路径参数

| 参数  | 是否必选 | 参数类型   | 描述  |
|-----|------|--------|-----|
| key | 是    | String | 标签键 |

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                         |
|---------------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| project_id    | 是    | String | 项目ID                                                                                                                                                                                                                                                                                                                       |
| resource_id   | 是    | String | 资源ID                                                                                                                                                                                                                                                                                                                       |
| resource_type | 是    | String | 标签资源类型: <ul style="list-style-type: none"><li>• instance: 企业路由器实例</li><li>• route-table: 路由表</li><li>• vpc-attachment: VPC连接</li><li>• vgw-attachment: 虚拟网关连接</li><li>• peering-attachment: 对等连接（Peering）连接</li><li>• vpn-attachment: VPN网关连接</li><li>• ecn-attachment: 企业连接网连接</li><li>• cfw-attachment: 云防火墙连接</li></ul> |

请求参数

无

响应参数

无

请求示例

```
DELETE https://{erouter-endpoint}/v3/cfa563efb77d4b6d9960781d82530fd8/instance/3320166e-b937-40cc-a35c-02cd3f2b3ee2/tags/key1
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DeleteResourceTagSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteResourceTagRequest request = new DeleteResourceTagRequest();
        request.withKey("{key}");
        request.withResourceId("{resource_id}");

        request.withResourceType(DeleteResourceTagRequest.ResourceTypeEnum.fromValue("{resource_type}"));
        try {
            DeleteResourceTagResponse response = client.deleteResourceTag(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)
```

```
client = ErClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(ErRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = DeleteResourceTagRequest()
    request.key = "{key}"
    request.resource_id = "{resource_id}"
    request.resource_type = "{resource_type}"
    response = client.delete_resource_tag(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteResourceTagRequest{}
    request.Key = "{key}"
    request.ResourceId = "{resource_id}"
    request.ResourceType = model.GetDeleteResourceTagRequestResourceTypeEnum().RESOURCE_TYPE
    response, err := client.DeleteResourceTag(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述         |
|-----|------------|
| 204 | No Content |

## 错误码

请参见[错误码](#)。

# 4.9 配额管理

## 4.9.1 查询配额

### 功能介绍

查询租户各类资源的使用情况，如企业路由器的使用量，VPC连接的使用量等。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v3/{project\_id}/enterprise-router/quotas

表 4-186 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

表 4-187 Query 参数

| 参数    | 是否必选 | 参数类型    | 描述                                                            |
|-------|------|---------|---------------------------------------------------------------|
| limit | 否    | Integer | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b> |

| 参数           | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------|------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| marker       | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度：1<br>最大长度：128                                                                                                                                                                                                                                                                                                                                                                         |
| type         | 否    | Array of strings | 支持过滤的配额类型： <ul style="list-style-type: none"><li>er_instance: 企业路由器实例的配额和使用量</li><li>dc_attachment: 云专线网关连接的配额和使用量</li><li>vpc_attachment: VPC连接的配额和使用量</li><li>vpn_attachment: VPN网关连接的配额和使用量</li><li>peering_attachment: 云连接实例连接的配额和使用量</li><li>can_attachment: 智能接入网关连接的配额和使用量</li><li>route_table: 路由表的配额和使用量</li><li>static_route: 静态路由的配额和使用量</li><li>vpc_er: 每个vpc可以接入的企业路由器数量和当前使用量</li><li>flow_log: 每个连接可以创建的流日志数量</li></ul> |
| erId         | 否    | Object           | 企业路由器ID                                                                                                                                                                                                                                                                                                                                                                                                                                |
| routeTableId | 否    | Object           | 路由表ID                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| vpcId        | 否    | Object           | VPC ID                                                                                                                                                                                                                                                                                                                                                                                                                                 |

请求参数

无

响应参数

状态码：200

表 4-188 响应 Body 参数

| 参数        | 参数类型                                   | 描述      |
|-----------|----------------------------------------|---------|
| quotas    | Array of <a href="#">Quota</a> objects | 配额使用详情  |
| page_info | <a href="#">PageInfo</a> object        | 分页查询的信息 |

表 4-189 Quota

| 参数          | 参数类型   | 描述               |
|-------------|--------|------------------|
| quota_key   | String | 配额类型             |
| quota_limit | Long   | 可用的配额数，-1 代表不受限制 |
| used        | Long   | 已使用的配额数量         |
| unit        | String | 用量单位             |

表 4-190 PageInfo

| 参数            | 参数类型    | 描述                    |
|---------------|---------|-----------------------|
| next_marker   | String  | 下一个marker标识，为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量             |

请求示例

查询配额信息

GET https://{erouter\_endpoint}/v3/08d5a9564a704afda6039ae2babbe3c/enterprise-router/quotas

响应示例

状态码：200

OK

```
{
  "quotas": [ {
    "quota_key": "er_instance",
    "quota_limit": 1,
    "used": 0,
    "unit": "count"
  }, {
    "quota_key": "dc_attachment",
    "quota_limit": 2,
    "used": 0,
    "unit": "count"
  }, {
    "quota_key": "route_table",
```

```
"quota_limit" : 20,
"used" : 5,
"unit" : "count"
}, {
  "quota_key" : "static_route",
  "quota_limit" : 500,
  "used" : 2,
  "unit" : "count"
}, {
  "quota_key" : "can_attachment",
  "quota_limit" : 10,
  "used" : 0,
  "unit" : "count"
}, {
  "quota_key" : "connect_attachment",
  "quota_limit" : 20,
  "used" : 0,
  "unit" : "count"
}, {
  "quota_key" : "peering_attachment",
  "quota_limit" : 10,
  "used" : 0,
  "unit" : "count"
}, {
  "quota_key" : "vpn_attachment",
  "quota_limit" : 10,
  "used" : 0,
  "unit" : "count"
}, {
  "quota_key" : "flow_log",
  "quota_limit" : 20,
  "used" : 4,
  "unit" : "count"
}],
"page_info" : {
  "next_marker" : "1",
  "current_count" : 9
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ShowQuotasSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
String projectId = "{project_id}";

ICredential auth = new BasicCredentials()
    .withProjectId(projectId)
    .withAk(ak)
    .withSk(sk);

ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
ShowQuotasRequest request = new ShowQuotasRequest();
try {
    ShowQuotasResponse response = client.showQuotas(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowQuotasRequest()
        response = client.show_quotas(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowQuotasRequest{}
    response, err := client.ShowQuotas(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.10 可用区

## 4.10.1 查询可用区列表

### 功能介绍

查询支持创建企业路由器实例的可用区列表，当可用区状态为available时，表示可以创建企业路由器实例。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/availability-zones

表 4-191 路径参数

| 参数         | 是否必选 | 参数类型   | 描述   |
|------------|------|--------|------|
| project_id | 是    | String | 项目ID |

表 4-192 Query 参数

| 参数          | 是否必选 | 参数类型   | 描述      |
|-------------|------|--------|---------|
| instance_id | 否    | String | 企业路由器ID |

### 请求参数

无

### 响应参数

状态码：200

表 4-193 响应 Body 参数

| 参数                 | 参数类型                                           | 描述    |
|--------------------|------------------------------------------------|-------|
| availability_zones | Array of <a href="#">AvailableZone</a> objects | 可用区列表 |
| request_id         | String                                         | 请求ID  |

表 4-194 AvailableZone

| 参数    | 参数类型   | 描述                           |
|-------|--------|------------------------------|
| code  | String | 可用区编码                        |
| state | String | 可用区是否可用available unavailable |

请求示例

查询可用区列表

GET https://{erouter\_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/availability-zones

响应示例

状态码：200

OK

```
{
  "availability_zones": [ {
    "code": "az1",
    "state": "available"
  }, {
    "code": "az2",
    "state": "unavailable"
  } ]
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class ListAvailabilityZoneSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
```

```
.withProjectId(projectId)
.withAk(ak)
.withSk(sk);

ErClient client = ErClient.newBuilder()
    .withCredential(auth)
    .withRegion(ErRegion.valueOf("<YOUR REGION>"))
    .build();
ListAvailabilityZoneRequest request = new ListAvailabilityZoneRequest();
try {
    ListAvailabilityZoneResponse response = client.listAvailabilityZone(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsaker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsaker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListAvailabilityZoneRequest()
        response = client.list_availability_zone(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListAvailabilityZoneRequest{}
    response, err := client.ListAvailabilityZone(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.11 流日志

### 4.11.1 创建流日志

#### 功能介绍

给ER实例创建流日志。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs

表 4-195 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

#### 请求参数

表 4-196 请求 Header 参数

| 参数             | 是否必选 | 参数类型   | 描述                         |
|----------------|------|--------|----------------------------|
| X-Client-Token | 否    | String | 幂等性标识<br>最小长度：1<br>最大长度：64 |

表 4-197 请求 Body 参数

| 参数       | 是否必选 | 参数类型                                  | 描述       |
|----------|------|---------------------------------------|----------|
| flow_log | 是    | <a href="#">FlowLogRequest</a> object | 创建流日志请求体 |

表 4-198 FlowLogRequest

| 参数   | 是否必选 | 参数类型   | 描述                         |
|------|------|--------|----------------------------|
| name | 是    | String | 流日志名称<br>最小长度：1<br>最大长度：64 |

| 参数             | 是否必选 | 参数类型   | 描述                                                                                                  |
|----------------|------|--------|-----------------------------------------------------------------------------------------------------|
| description    | 否    | String | 流日志描述<br>最小长度：0<br>最大长度：255                                                                         |
| resource_type  | 是    | String | 流日志采集的资源类型： <ul style="list-style-type: none"><li>• VPC连接</li><li>• 虚拟网关连接</li><li>• 对等连接</li></ul> |
| resource_id    | 是    | String | 要采集的资源ID                                                                                            |
| log_group_id   | 是    | String | 日志组ID。请在云日志服务中获取，详情请参见《云日志服务用户指南》。                                                                  |
| log_stream_id  | 是    | String | 日志主题ID。请在云日志服务中获取，详情请参见《云日志服务用户指南》。                                                                 |
| log_store_type | 是    | String | 流日志的存储类型： <ul style="list-style-type: none"><li>• LTS: 云日志服务器存储</li></ul>                           |

响应参数

状态码：202

表 4-199 响应 Header 参数

| 参数             | 参数类型   | 描述    |
|----------------|--------|-------|
| X-Client-Token | String | 幂等性标识 |

表 4-200 响应 Body 参数

| 参数         | 参数类型                  | 描述    |
|------------|-----------------------|-------|
| flow_log   | <b>FlowLog</b> object | 流日志详情 |
| request_id | String                | 请求id  |

表 4-201 FlowLog

| 参数 | 参数类型   | 描述    |
|----|--------|-------|
| id | String | 流日志ID |

| 参数                       | 参数类型    | 描述                                                         |
|--------------------------|---------|------------------------------------------------------------|
| name                     | String  | 流日志名称                                                      |
| description              | String  | 流日志描述                                                      |
| project_id               | String  | 流日志任务创建者项目ID                                               |
| resource_type            | String  | 采集的资源类型:attachment                                         |
| resource_id              | String  | 采集的资源ID                                                    |
| log_group_id             | String  | 日志组ID                                                      |
| log_stream_id            | String  | 日志流ID                                                      |
| log_store_type           | String  | 流日志存储类型:LTS                                                |
| log_aggregation_interval | Integer | 日志聚合时间，单位s，取值范围：60-600                                     |
| created_at               | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| updated_at               | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| state                    | String  | 日志资源状态:pending available modifying deleting deleted failed |
| enabled                  | Boolean | 日志开关:true false                                            |

请求示例

创建流日志

```
POST https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/a43c55e9-4911-4030-90e1-5c2bf6ae6fe2/flow-logs
{
  "flow_log": {
    "name": "flow-log",
    "description": "flow log test",
    "resource_type": "attachment",
    "resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",
    "log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",
    "log_store_type": "LTS"
  }
}
```

响应示例

状态码：202

Accepted

```
{
  "flow_log": {
    "id": "b216bc1d-5963-41a7-89f9-779a5128c5ac",
  }
}
```

```
"name": "flow_log",
"project_id": "0605767a9980d5762fbcc00b3537e757",
"resource_type": "attachment",
"resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",
"log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",
"log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",
"log_store_type": "LTS",
"log_aggregation_interval": 600,
"created_at": "2020-03-11T15:13:31Z",
"updated_at": "2022-03-11T15:13:31Z",
"state": "available",
"enabled": true,
"description": "ER流日志"
},
"request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

#### 创建流日志

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class CreateFlowLogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
            .build();
        CreateFlowLogRequest request = new CreateFlowLogRequest();
        request.withErId("{er_id}");
        CreateFlowLogRequestBody body = new CreateFlowLogRequestBody();
        FlowLogRequest flowLogbody = new FlowLogRequest();
        flowLogbody.withName("flow-log")
            .withDescription("flow log test")
            .withResourceType(FlowLogRequest.ResourceTypeEnum.fromValue("attachment"))
            .withResourceId("6f83b848-8331-4271-ac0c-ef94b7686402")
            .withLogGroupId("0139393c-eeb2-49f0-bbd4-c5faec6b1497")
    }
```

```
.withLogStreamId("d22c3b44-2f71-470f-83f3-96a8af6956ad")
.withLogStoreType(FlowLogRequest.LogStoreTypeEnum.fromValue("LTS"));
body.withFlowLog(flowLogbody);
request.withBody(body);
try {
    CreateFlowLogResponse response = client.createFlowLog(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrMsg());
}
}
```

## Python

### 创建流日志

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateFlowLogRequest()
        request.er_id = "{er_id}"
        flowLogbody = FlowLogRequest(
            name="flow-log",
            description="flow log test",
            resource_type="attachment",
            resource_id="6f83b848-8331-4271-ac0c-ef94b7686402",
            log_group_id="0139393c-eeb2-49f0-bbd4-c5faec6b1497",
            log_stream_id="d22c3b44-2f71-470f-83f3-96a8af6956ad",
            log_store_type="LTS"
        )
        request.body = CreateFlowLogRequestBody(
            flow_log=flowLogbody
        )
        response = client.create_flow_log(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

### 创建流日志

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateFlowLogRequest{}
    request.ErId = "{er_id}"
    descriptionFlowLog := "flow log test"
    flowLogbody := &model.FlowLogRequest{
        Name: "flow-log",
        Description: &descriptionFlowLog,
        ResourceType: model.GetFlowLogRequestResourceTypeEnum().ATTACHMENT,
        ResourceId: "6f83b848-8331-4271-ac0c-ef94b7686402",
        LogGroupId: "0139393c-eeb2-49f0-bbd4-c5faec6b1497",
        LogStreamId: "d22c3b44-2f71-470f-83f3-96a8af6956ad",
        LogStoreType: model.GetFlowLogRequestLogStoreTypeEnum().LTS,
    }
    request.Body = &model.CreateFlowLogRequestBody{
        FlowLog: flowLogbody,
    }
    response, err := client.CreateFlowLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

错误码

请参见[错误码](#)。

4.11.2 查询流日志列表

功能介绍

查询企业路由器实例下的流日志列表

接口约束

- 支持过滤查询，过滤条件有resource\_type，resource\_id，enterprise\_project\_id过滤条件可以重复和组合。
- 支持分页查询，limit和marker组合实现分页查询。
- 支持单字段排序功能，排序字段有[id,name,state]，不支持多字段排序。

调用方法

请参见[如何调用API](#)。

URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs

表 4-202 路径参数

| 参数         | 是否必选 | 参数类型   | 描述        |
|------------|------|--------|-----------|
| project_id | 是    | String | 项目ID      |
| er_id      | 是    | String | 企业路由器实例ID |

表 4-203 Query 参数

| 参数            | 是否必选 | 参数类型             | 描述          |
|---------------|------|------------------|-------------|
| resource_type | 否    | String           | 采集的资源类型     |
| resource_id   | 否    | Array of strings | 连接对应的资源ID列表 |

| 参数       | 是否必选 | 参数类型             | 描述                                                                             |
|----------|------|------------------|--------------------------------------------------------------------------------|
| limit    | 否    | Integer          | 每页返回的个数。<br>取值范围：0~2000。<br>最小值： <b>0</b><br>最大值： <b>2000</b>                  |
| marker   | 否    | String           | 查询的页码id，为空时为查询第一页。<br>使用说明：必须与limit一起使用。<br>最小长度： <b>1</b><br>最大长度： <b>128</b> |
| sort_key | 否    | Array of strings | 按关键字排序，默认按照id排序，可选值:id name state                                              |
| sort_dir | 否    | Array of strings | 返回结果按照升序或降序排列，默认为asc,降序为desc                                                   |

请求参数

无

响应参数

状态码：200

表 4-204 响应 Body 参数

| 参数          | 参数类型                            | 描述      |
|-------------|---------------------------------|---------|
| flow_logs   | Array of <b>FlowLog</b> objects | 流日志详情   |
| request_id  | String                          | 请求ID    |
| total_count | Long                            | 总计数量    |
| page_info   | <b>PageInfo</b> object          | 分页查询的信息 |

表 4-205 FlowLog

| 参数          | 参数类型   | 描述    |
|-------------|--------|-------|
| id          | String | 流日志ID |
| name        | String | 流日志名称 |
| description | String | 流日志描述 |

| 参数                       | 参数类型    | 描述                                                         |
|--------------------------|---------|------------------------------------------------------------|
| project_id               | String  | 流日志任务创建者项目ID                                               |
| resource_type            | String  | 采集的资源类型:attachment                                         |
| resource_id              | String  | 采集的资源ID                                                    |
| log_group_id             | String  | 日志组ID                                                      |
| log_stream_id            | String  | 日志流ID                                                      |
| log_store_type           | String  | 流日志存储类型:LTS                                                |
| log_aggregation_interval | Integer | 日志聚合时间, 单位s, 取值范围: 60-600                                  |
| created_at               | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| updated_at               | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| state                    | String  | 日志资源状态:pending available modifying deleting deleted failed |
| enabled                  | Boolean | 日志开关:true false                                            |

表 4-206 PageInfo

| 参数            | 参数类型    | 描述                     |
|---------------|---------|------------------------|
| next_marker   | String  | 下一个marker标识, 为空时表示最后一页 |
| current_count | Integer | 当前列表中资源数量              |

请求示例

查询流日志列表

```
GET https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/a43c55e9-4911-4030-90e1-5c2bf6ae6fe2/flow-logs
```

响应示例

状态码: 200

OK

```
{
  "flow_logs": [ {
    "id": "b216bc1d-5963-41a7-89f9-779a5128c5ac",
    "name": "flow_log",
    "project_id": "0605767a9980d5762fbcc00b3537e757",
    "resource_type": "attachment",
```

```
{
  "resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",
  "log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",
  "log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",
  "log_store_type": "LTS",
  "log_aggregation_interval": 600,
  "created_at": "2020-03-11T15:13:31Z",
  "updated_at": "2022-03-11T15:13:31Z",
  "state": "available",
  "enabled": true,
  "description": "ER流日志"
},
{
  "page_info": {
    "next_marker": "1",
    "current_count": 1
  },
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

## 错误码

请参见[错误码](#)。

## 4.11.3 查询流日志详情

### 功能介绍

查询流日志详情

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs/{flow\_log\_id}

表 4-207 路径参数

| 参数          | 是否必选 | 参数类型   | 描述        |
|-------------|------|--------|-----------|
| project_id  | 是    | String | 项目ID      |
| er_id       | 是    | String | 企业路由器实例ID |
| flow_log_id | 是    | String | 流日志ID     |

请求参数

无

响应参数

状态码：200

表 4-208 响应 Body 参数

| 参数         | 参数类型           | 描述    |
|------------|----------------|-------|
| flow_log   | FlowLog object | 流日志详情 |
| request_id | String         | 请求id  |

表 4-209 FlowLog

| 参数                       | 参数类型    | 描述                                                         |
|--------------------------|---------|------------------------------------------------------------|
| id                       | String  | 流日志ID                                                      |
| name                     | String  | 流日志名称                                                      |
| description              | String  | 流日志描述                                                      |
| project_id               | String  | 流日志任务创建者项目ID                                               |
| resource_type            | String  | 采集的资源类型:attachment                                         |
| resource_id              | String  | 采集的资源ID                                                    |
| log_group_id             | String  | 日志组ID                                                      |
| log_stream_id            | String  | 日志流ID                                                      |
| log_store_type           | String  | 流日志存储类型:LTS                                                |
| log_aggregation_interval | Integer | 日志聚合时间，单位s，取值范围：60-600                                     |
| created_at               | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| updated_at               | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| state                    | String  | 日志资源状态:pending available modifying deleting deleted failed |
| enabled                  | Boolean | 日志开关:true false                                            |

请求示例

查询流日志列表

```
GET https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/  
a43c55e9-4911-4030-90e1-5c2bf6ae6fe2/flow-logs/b216bc1d-5963-41a7-89f9-779a5128c5ac
```

## 响应示例

状态码：200

OK

```
{  
  "flow_log": {  
    "id": "b216bc1d-5963-41a7-89f9-779a5128c5ac",  
    "name": "flow_log",  
    "project_id": "0605767a9980d5762fbcc00b3537e757",  
    "resource_type": "attachment",  
    "resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",  
    "log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",  
    "log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",  
    "log_store_type": "LTS",  
    "log_aggregation_interval": 600,  
    "created_at": "2020-03-11T15:13:31Z",  
    "updated_at": "2022-03-11T15:13:31Z",  
    "state": "available",  
    "enabled": true,  
    "description": "ER流日志"  
  },  
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.er.v3.region.ErRegion;  
import com.huaweicloud.sdk.er.v3.*;  
import com.huaweicloud.sdk.er.v3.model.*;  
  
public class ShowFlowLogSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        ErClient client = ErClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))  
    }  
}
```

```
        .build();
        ShowFlowLogRequest request = new ShowFlowLogRequest();
        request.withFlowLogId("{flow_log_id}");
        request.withErId("{er_id}");
        try {
            ShowFlowLogResponse response = client.showFlowLog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsddker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsddker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowFlowLogRequest()
        request.flow_log_id = "{flow_log_id}"
        request.er_id = "{er_id}"
        response = client.show_flow_log(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowFlowLogRequest{}
    request.FlowLogId = "{flow_log_id}"
    request.ErId = "{er_id}"
    response, err := client.ShowFlowLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.11.4 更新流日志基本信息

功能介绍

更新流日志基本信息

调用方法

请参见[如何调用API](#)。

URI

PUT /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs/{flow\_log\_id}

表 4-210 路径参数

| 参数          | 是否必选 | 参数类型   | 描述        |
|-------------|------|--------|-----------|
| project_id  | 是    | String | 项目ID      |
| er_id       | 是    | String | 企业路由器实例ID |
| flow_log_id | 是    | String | 流日志ID     |

请求参数

表 4-211 请求 Body 参数

| 参数          | 是否必选 | 参数类型   | 描述                         |
|-------------|------|--------|----------------------------|
| name        | 否    | String | 流日志名称<br>最小长度：1<br>最大长度：64 |
| description | 否    | String | 最小长度：0<br>最大长度：255         |

响应参数

状态码：200

表 4-212 响应 Body 参数

| 参数         | 参数类型           | 描述    |
|------------|----------------|-------|
| flow_log   | FlowLog object | 流日志详情 |
| request_id | String         | 请求id  |

表 4-213 FlowLog

| 参数          | 参数类型   | 描述    |
|-------------|--------|-------|
| id          | String | 流日志ID |
| name        | String | 流日志名称 |
| description | String | 流日志描述 |

| 参数                       | 参数类型    | 描述                                                         |
|--------------------------|---------|------------------------------------------------------------|
| project_id               | String  | 流日志任务创建者项目ID                                               |
| resource_type            | String  | 采集的资源类型:attachment                                         |
| resource_id              | String  | 采集的资源ID                                                    |
| log_group_id             | String  | 日志组ID                                                      |
| log_stream_id            | String  | 日志流ID                                                      |
| log_store_type           | String  | 流日志存储类型:LTS                                                |
| log_aggregation_interval | Integer | 日志聚合时间, 单位s, 取值范围: 60-600                                  |
| created_at               | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| updated_at               | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| state                    | String  | 日志资源状态:pending available modifying deleting deleted failed |
| enabled                  | Boolean | 日志开关:true false                                            |

## 请求示例

更新流日志b216bc1d-5963-41a7-89f9-779a5128c5ac的名称为flow\_log\_update,描述为flow log update test

```
PUT https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/6f83b848-8331-4271-ac0c-ef94b7686402/flow_logs/b216bc1d-5963-41a7-89f9-779a5128c5ac
{
  "name": "flow_log_update",
  "description": "flow log update test"
}
```

## 响应示例

状态码: 200

OK

```
{
  "flow_log": {
    "id": "b216bc1d-5963-41a7-89f9-779a5128c5ac",
    "name": "flow_log_update",
    "project_id": "0605767a9980d5762fbcc00b3537e757",
    "resource_type": "attachment",
    "resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",
    "log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",
    "log_store_type": "LTS",
    "log_aggregation_interval": 600,
    "created_at": "2020-03-11T15:13:31Z",
    "updated_at": "2022-03-11T15:13:31Z",
  }
}
```

```
"state" : "available",
"enabled" : true,
"description" : "flow log update test"
},
"request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新流日志b216bc1d-5963-41a7-89f9-779a5128c5ac的名称为flow\_log\_update,描述为flow log update test

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class UpdateFlowLogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateFlowLogRequest request = new UpdateFlowLogRequest();
        request.withFlowLogId("{flow_log_id}");
        request.withErId("{er_id}");
        UpdateFlowLogRequestBody body = new UpdateFlowLogRequestBody();
        body.withDescription("flow log update test");
        body.withName("flow_log_update");
        request.withBody(body);
        try {
            UpdateFlowLogResponse response = client.updateFlowLog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
        }
    }
}
```

```
        System.out.println(e.getErrorCode());  
        System.out.println(e.getErrorMsg());  
    }  
}  
}
```

## Python

更新流日志b216bc1d-5963-41a7-89f9-779a5128c5ac的名称为flow\_log\_update,描述为flow log update test

```
# coding: utf-8
```

```
import os  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdker.v3.region.er_region import ErRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdker.v3 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    # variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = os.environ["CLOUD_SDK_AK"]  
    sk = os.environ["CLOUD_SDK_SK"]  
    projectId = "{project_id}"  
  
    credentials = BasicCredentials(ak, sk, projectId)  
  
    client = ErClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(ErRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = UpdateFlowLogRequest()  
        request.flow_log_id = "{flow_log_id}"  
        request.er_id = "{er_id}"  
        request.body = UpdateFlowLogRequestBody(  
            description="flow log update test",  
            name="flow_log_update"  
        )  
        response = client.update_flow_log(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

## Go

更新流日志b216bc1d-5963-41a7-89f9-779a5128c5ac的名称为flow\_log\_update,描述为flow log update test

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"  
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateFlowLogRequest{}
    request.FlowLogId = "{flow_log_id}"
    request.ErId = "{er_id}"
    descriptionUpdateFlowLogRequestBody:= "flow log update test"
    nameUpdateFlowLogRequestBody:= "flow_log_update"
    request.Body = &model.UpdateFlowLogRequestBody{
        Description: &descriptionUpdateFlowLogRequestBody,
        Name: &nameUpdateFlowLogRequestBody,
    }
    response, err := client.UpdateFlowLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述 |
|-----|----|
| 200 | OK |

错误码

请参见[错误码](#)。

4.11.5 删除流日志

功能介绍

删除流日志

## 接口约束

流日志状态为available，deleting和failed的时候才能删除。

## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs/{flow\_log\_id}

表 4-214 路径参数

| 参数          | 是否必选 | 参数类型   | 描述        |
|-------------|------|--------|-----------|
| project_id  | 是    | String | 项目ID      |
| er_id       | 是    | String | 企业路由器实例ID |
| flow_log_id | 是    | String | 流日志ID     |

## 请求参数

无

## 响应参数

无

## 请求示例

删除流日志

```
DELETE https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/6f83b848-8331-4271-ac0c-ef94b7686402/flow_logs/b216bc1d-5963-41a7-89f9-779a5128c5ac
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
```

```
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class DeleteFlowLogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteFlowLogRequest request = new DeleteFlowLogRequest();
        request.withFlowLogId("{flow_log_id}");
        request.withErId("{er_id}");
        try {
            DeleteFlowLogResponse response = client.deleteFlowLog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.getenv("CLOUD_SDK_AK")
    sk = os.getenv("CLOUD_SDK_SK")
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(ErRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = DeleteFlowLogRequest()
    request.flow_log_id = "{flow_log_id}"
    request.er_id = "{er_id}"
    response = client.delete_flow_log(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteFlowLogRequest{}
    request.FlowLogId = "{flow_log_id}"
    request.ErId = "{er_id}"
    response, err := client.DeleteFlowLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

## 4.11.6 开启流日志

### 功能介绍

开启流日志

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs/{flow\_log\_id}/enable

表 4-215 路径参数

| 参数          | 是否必选 | 参数类型   | 描述        |
|-------------|------|--------|-----------|
| project_id  | 是    | String | 项目ID      |
| er_id       | 是    | String | 企业路由器实例ID |
| flow_log_id | 是    | String | 流日志ID     |

### 请求参数

无

### 响应参数

状态码：202

表 4-216 响应 Body 参数

| 参数         | 参数类型                           | 描述    |
|------------|--------------------------------|-------|
| flow_log   | <a href="#">FlowLog</a> object | 流日志详情 |
| request_id | String                         | 请求id  |

表 4-217 FlowLog

| 参数                       | 参数类型    | 描述                                                         |
|--------------------------|---------|------------------------------------------------------------|
| id                       | String  | 流日志ID                                                      |
| name                     | String  | 流日志名称                                                      |
| description              | String  | 流日志描述                                                      |
| project_id               | String  | 流日志任务创建者项目ID                                               |
| resource_type            | String  | 采集的资源类型:attachment                                         |
| resource_id              | String  | 采集的资源ID                                                    |
| log_group_id             | String  | 日志组ID                                                      |
| log_stream_id            | String  | 日志流ID                                                      |
| log_store_type           | String  | 流日志存储类型:LTS                                                |
| log_aggregation_interval | Integer | 日志聚合时间, 单位s, 取值范围: 60-600                                  |
| created_at               | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| updated_at               | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| state                    | String  | 日志资源状态:pending available modifying deleting deleted failed |
| enabled                  | Boolean | 日志开关:true false                                            |

请求示例

开启流日志

```
POST https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/a43c55e9-4911-4030-90e1-5c2bf6ae6fe2/flow-logs/b216bc1d-5963-41a7-89f9-779a5128c5ac/enable
```

响应示例

状态码: 202

Accepted

```
{
  "flow_log": {
    "id": "b216bc1d-5963-41a7-89f9-779a5128c5ac",
    "name": "flow_log_update",
    "project_id": "0605767a9980d5762fbcc00b3537e757",
    "resource_type": "attachment",
    "resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",
    "log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",
    "log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",
    "log_store_type": "LTS",
    "log_aggregation_interval": 600,
  }
}
```

```
"created_at" : "2020-03-11T15:13:31Z",
"updated_at" : "2022-03-11T15:13:31Z",
"state" : "available",
"enabled" : true,
"description" : "ER流日志"
},
"request_id" : "915a14a6-867b-4af7-83d1-70efceb146f9"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.er.v3.region.ErRegion;
import com.huaweicloud.sdk.er.v3.*;
import com.huaweicloud.sdk.er.v3.model.*;

public class EnableFlowLogSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
        String projectId = "{project_id}";

        ICredential auth = new BasicCredentials()
            .withProjectId(projectId)
            .withAk(ak)
            .withSk(sk);

        ErClient client = ErClient.newBuilder()
            .withCredential(auth)
            .withRegion(ErRegion.valueOf("<YOUR REGION>"))
            .build();
        EnableFlowLogRequest request = new EnableFlowLogRequest();
        request.withFlowLogId("{flow_log_id}");
        request.withErId("{er_id}");
        try {
            EnableFlowLogResponse response = client.enableFlowLog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = EnableFlowLogRequest()
        request.flow_log_id = "{flow_log_id}"
        request.er_id = "{er_id}"
        response = client.enable_flow_log(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).
WithCredential(auth).
Build())

request := &model.EnableFlowLogRequest{}
request.FlowLogId = "{flow_log_id}"
request.ErId = "{er_id}"
response, err := client.EnableFlowLog(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

错误码

请参见[错误码](#)。

4.11.7 关闭流日志

功能介绍

关闭流日志

调用方法

请参见[如何调用API](#)。

URI

POST /v3/{project\_id}/enterprise-router/{er\_id}/flow-logs/{flow\_log\_id}/disable

表 4-218 路径参数

| 参数          | 是否必选 | 参数类型   | 描述        |
|-------------|------|--------|-----------|
| project_id  | 是    | String | 项目ID      |
| er_id       | 是    | String | 企业路由器实例ID |
| flow_log_id | 是    | String | 流日志ID     |

请求参数

无

响应参数

状态码：202

表 4-219 响应 Body 参数

| 参数         | 参数类型           | 描述    |
|------------|----------------|-------|
| flow_log   | FlowLog object | 流日志详情 |
| request_id | String         | 请求id  |

表 4-220 FlowLog

| 参数                       | 参数类型    | 描述                                                         |
|--------------------------|---------|------------------------------------------------------------|
| id                       | String  | 流日志ID                                                      |
| name                     | String  | 流日志名称                                                      |
| description              | String  | 流日志描述                                                      |
| project_id               | String  | 流日志任务创建者项目ID                                               |
| resource_type            | String  | 采集的资源类型:attachment                                         |
| resource_id              | String  | 采集的资源ID                                                    |
| log_group_id             | String  | 日志组ID                                                      |
| log_stream_id            | String  | 日志流ID                                                      |
| log_store_type           | String  | 流日志存储类型:LTS                                                |
| log_aggregation_interval | Integer | 日志聚合时间，单位s，取值范围：60-600                                     |
| created_at               | String  | 创建时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| updated_at               | String  | 更新时间,格式:YYYY-MM-DDTHH:mm:ss.sssZ                           |
| state                    | String  | 日志资源状态:pending available modifying deleting deleted failed |
| enabled                  | Boolean | 日志开关:true false                                            |

请求示例

关闭流日志

```
POST https://{erouter_endpoint}/v3/0605767a9980d5762fbcc00b3537e757/enterprise-router/  
a43c55e9-4911-4030-90e1-5c2bf6ae6fe2/flow-logs/b216bc1d-5963-41a7-89f9-779a5128c5ac/disable
```

## 响应示例

状态码：202

Accepted

```
{  
  "flow_log": {  
    "id": "b216bc1d-5963-41a7-89f9-779a5128c5ac",  
    "name": "flow_log_update",  
    "project_id": "0605767a9980d5762fbcc00b3537e757",  
    "resource_type": "attachment",  
    "resource_id": "6f83b848-8331-4271-ac0c-ef94b7686402",  
    "log_group_id": "0139393c-eeb2-49f0-bbd4-c5faec6b1497",  
    "log_stream_id": "d22c3b44-2f71-470f-83f3-96a8af6956ad",  
    "log_store_type": "LTS",  
    "log_aggregation_interval": 600,  
    "created_at": "2020-03-11T15:13:31Z",  
    "updated_at": "2022-03-11T15:13:31Z",  
    "state": "available",  
    "enabled": true,  
    "description": "ER流日志"  
  },  
  "request_id": "915a14a6-867b-4af7-83d1-70efceb146f9"  
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.er.v3.region.ErRegion;  
import com.huaweicloud.sdk.er.v3.*;  
import com.huaweicloud.sdk.er.v3.model.*;  
  
public class DisableFlowLogSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
        String projectId = "{project_id}";  
  
        ICredential auth = new BasicCredentials()  
            .withProjectId(projectId)  
            .withAk(ak)  
            .withSk(sk);  
  
        ErClient client = ErClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(ErRegion.valueOf("<YOUR_REGION>"))
```

```
        .build();
        DisableFlowLogRequest request = new DisableFlowLogRequest();
        request.withFlowLogId("{flow_log_id}");
        request.withErId("{er_id}");
        try {
            DisableFlowLogResponse response = client.disableFlowLog(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrMsg());
        }
    }
}
```

## Python

```
# coding: utf-8

import os
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdsdker.v3.region.er_region import ErRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdsdker.v3 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    projectId = "{project_id}"

    credentials = BasicCredentials(ak, sk, projectId)

    client = ErClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(ErRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DisableFlowLogRequest()
        request.flow_log_id = "{flow_log_id}"
        request.er_id = "{er_id}"
        response = client.disable_flow_log(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

## Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    er "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/model"
```

```
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/er/v3/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")
    projectId := "{project_id}"

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        WithProjectId(projectId).
        Build()

    client := er.NewErClient(
        er.ErClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DisableFlowLogRequest{}
    request.FlowLogId = "{flow_log_id}"
    request.ErId = "{er_id}"
    response, err := client.DisableFlowLog(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述       |
|-----|----------|
| 202 | Accepted |

## 错误码

请参见[错误码](#)。

# 5 权限和授权项

## 5.1 权限及授权项说明

如果您需要对您所拥有的ER进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果华为账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用ER服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

权限根据授权的精细程度，分为**角色**和**策略**。角色以服务为粒度，是IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

### 说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来查询企业路由器列表，那么这个IAM用户被授予的策略中必须包含允许“er:instances:list”的授权项，该接口才能调用成功。

## 5.2 基于角色授权的服务授权项

本章节介绍企业路由器ER基于角色授权场景下支持的策略授权项。

### 支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：允许或拒绝对指定资源在特定条件下进行某项操作。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- 依赖的授权项：部分Action存在对其他Action的依赖，需要将依赖的Action同时写入授权项，才能实现对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。管理员可以在授权项列表中查看授权项是否支持IAM项目或企业项目，“√”表示支持，“×”表示暂不支持。关于IAM项目与企业项目的区别，详情请参见：[IAM与企业管理的区别](#)。

ER的支持自定义策略授权项如下所示：

- [企业路由器](#)
- [VPC连接](#)
- [其他连接](#)
- [路由表](#)
- [关联](#)
- [传播](#)
- [路由](#)
- [流日志](#)
- [标签](#)
- [配额管理](#)

### 5.3 企业路由器

| 权限                | 对应API接口                                                                             | 授权项<br>( Action )       | IAM项<br>目<br>(Projec<br>t) | 企业项目<br>(Enterpri<br>se<br>Project) |
|-------------------|-------------------------------------------------------------------------------------|-------------------------|----------------------------|-------------------------------------|
| 创建企业<br>路由器       | POST /v3/{project_id}/<br>enterprise-router/<br>instances                           | er:instances:crea<br>te | √                          | √                                   |
| 更新企业<br>路由器       | PUT /v3/{project_id}/<br>enterprise-router/<br>instances/<br>{enterprise_router_id} | er:instances:upd<br>ate | √                          | √                                   |
| 查询企业<br>路由器详<br>情 | GET /v3/{project_id}/<br>enterprise-router/<br>instances/<br>{enterprise_router_id} | er:instances:get        | √                          | √                                   |

| 权限          | 对应API接口                                                                                               | 授权项<br>( Action )     | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|-------------|-------------------------------------------------------------------------------------------------------|-----------------------|------------------------|------------------------------|
| 查询企业路由器列表   | GET /v3/{project_id}/enterprise-router/instances                                                      | er:instances:list     | √                      | √                            |
| 更新企业路由器的可用区 | POST /v3/{project_id}/enterprise-router/instances/{enterprise_router_id}/change-availability-zone-ids | er:instances:changeAZ | √                      | √                            |
| 删除企业路由器     | DELETE /v3/{project_id}/enterprise-router/instances/{enterprise_router_id}                            | er:instances:delete   | √                      | √                            |

## 5.4 VPC 连接

| 权限          | 对应API接口                                                                               | 授权项<br>( Action )     | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|-------------|---------------------------------------------------------------------------------------|-----------------------|------------------------|------------------------------|
| 创建VPC连接     | POST /v3/{project_id}/enterprise-router/{er_id}/vpc-attachments                       | er:attachments:create | √                      | √                            |
| 更新VPC连接基本信息 | PUT /v3/{project_id}/enterprise-router/{er_id}/vpc-attachments/{vpc_attachment_id}    | er:attachments:update | √                      | √                            |
| 查询VPC连接详情   | GET /v3/{project_id}/enterprise-router/{er_id}/vpc-attachments/{vpc_attachment_id}    | er:attachments:get    | √                      | √                            |
| 查询VPC连接列表   | GET /v3/{project_id}/enterprise-router/{er_id}/vpc-attachments                        | er:attachments:list   | √                      | √                            |
| 删除VPC连接     | DELETE /v3/{project_id}/enterprise-router/{er_id}/vpc-attachments/{vpc_attachment_id} | er:attachments:delete | √                      | √                            |

## 5.5 其他连接

| 权限           | 对应API接口                                                                                        | 授权项<br>( Action )         | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|--------------|------------------------------------------------------------------------------------------------|---------------------------|------------------------|------------------------------|
| 更新连接<br>基本信息 | PUT /v3/{project_id}/<br>enterprise-router/{er_id}/<br>attachments/<br>{attachment_id}         | er:attachments:<br>update | √                      | √                            |
| 查询连接<br>详情   | GET /v3/{project_id}/<br>enterprise-router/{er_id}/<br>attachments/<br>{attachment_id}         | er:attachments:<br>get    | √                      | √                            |
| 查询连接<br>列表   | GET /v3/{project_id}/<br>enterprise-router/{er_id}/<br>attachments                             | er:attachments:l<br>ist   | √                      | √                            |
| 接受共享<br>连接创建 | POST /v3/{project_id}/<br>enterprise-router/{er_id}/<br>attachments/<br>{attachment_id}/accept | er:attachments:<br>accept | √                      | √                            |
| 拒绝共享<br>连接创建 | POST /v3/{project_id}/<br>enterprise-router/{er_id}/<br>attachments/<br>{attachment_id}/reject | er:attachments:r<br>eject | √                      | √                            |

## 5.6 路由表

| 权限          | 对应API接口                                                                                  | 授权项<br>( Action )         | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|-------------|------------------------------------------------------------------------------------------|---------------------------|------------------------|------------------------------|
| 创建路由<br>表   | POST /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables                     | er:routeTables:cr<br>eate | √                      | √                            |
| 更新路由<br>表信息 | PUT /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables/<br>{route_table_id} | er:routeTables:u<br>pdate | √                      | √                            |

| 权限          | 对应API接口                                                                                     | 授权项<br>( Action )         | IAM项<br>目<br>(Project<br>) | 企业项目<br>(Enterpri<br>se<br>Project) |
|-------------|---------------------------------------------------------------------------------------------|---------------------------|----------------------------|-------------------------------------|
| 查询路由<br>表详情 | GET /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables/<br>{route_table_id}    | er:routeTables:ge<br>t    | √                          | √                                   |
| 查询路由<br>表列表 | GET /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables                         | er:routeTables:li<br>st   | √                          | √                                   |
| 删除路由<br>表   | DELETE /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables/<br>{route_table_id} | er:routeTables:d<br>elete | √                          | √                                   |

## 5.7 关联

| 权限         | 对应API接口                                                                                                    | 授权项<br>( Action )                | IAM项<br>目<br>(Projec<br>t) | 企业项目<br>(Enterpri<br>se<br>Project) |
|------------|------------------------------------------------------------------------------------------------------------|----------------------------------|----------------------------|-------------------------------------|
| 创建关联       | POST /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables/<br>{route_table_id}/<br>associate    | er:associations:a<br>ssociate    | √                          | √                                   |
| 查询关联<br>列表 | GET /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables/<br>{route_table_id}/<br>associations  | er:associations:li<br>st         | √                          | √                                   |
| 删除关联       | POST /v3/{project_id}/<br>enterprise-router/{er_id}/<br>route-tables/<br>{route_table_id}/<br>disassociate | er:associations:d<br>isassociate | √                          | √                                   |

## 5.8 传播

| 权限     | 对应API接口                                                                                            | 授权项<br>( Action )       | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|--------|----------------------------------------------------------------------------------------------------|-------------------------|------------------------|------------------------------|
| 创建传播   | POST /v3/{project_id}/enterprise-router/{er_id}/route-tables/{route_table_id}/enable-propagations  | er:propagations:enable  | √                      | √                            |
| 查询传播列表 | GET /v3/{project_id}/enterprise-router/{er_id}/route-tables/{route_table_id}/propagations          | er:propagations:list    | √                      | √                            |
| 删除传播   | POST /v3/{project_id}/enterprise-router/{er_id}/route-tables/{route_table_id}/disable-propagations | er:propagations:disable | √                      | √                            |

## 5.9 路由

| 权限       | 对应API接口                                                                                       | 授权项<br>( Action ) | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|----------|-----------------------------------------------------------------------------------------------|-------------------|------------------------|------------------------------|
| 创建静态路由   | POST /v3/{project_id}/enterprise-router/route-tables/{route_table_id}/static-routes           | er:routes:create  | √                      | √                            |
| 修改静态路由   | PUT /v3/{project_id}/enterprise-router/route-tables/{route_table_id}/static-routes/{route_id} | er:routes:update  | √                      | √                            |
| 查询静态路由详情 | GET /v3/{project_id}/enterprise-router/route-tables/{route_table_id}/static-routes/{route_id} | er:routes:get     | √                      | √                            |

| 权限       | 对应API接口                                                                                          | 授权项<br>( Action ) | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|----------|--------------------------------------------------------------------------------------------------|-------------------|------------------------|------------------------------|
| 查询静态路由列表 | GET /v3/{project_id}/enterprise-router/route-tables/{route_table_id}/static-routes               | er:routes:list    | √                      | √                            |
| 查询有效路由列表 | GET /v3/{project_id}/enterprise-router/route-tables/{route_table_id}/routes                      | er:routes:list    | √                      | √                            |
| 删除静态路由   | DELETE /v3/{project_id}/enterprise-router/route-tables/{route_table_id}/static-routes/{route_id} | er:routes:delete  | √                      | √                            |

## 5.10 流日志

| 权限        | 对应API接口                                                                        | 授权项<br>( Action )  | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|-----------|--------------------------------------------------------------------------------|--------------------|------------------------|------------------------------|
| 创建流日志     | POST /v3/{project_id}/enterprise-router/{er_id}/flow-logs                      | er:flowlogs:create | √                      | √                            |
| 查询流日志列表   | GET /v3/{project_id}/enterprise-router/{er_id}/flow-logs                       | er:flowlogs:list   | √                      | √                            |
| 查询流日志     | GET /v3/{project_id}/enterprise-router/{er_id}/flow-logs/{flow_log_id}         | er:flowlogs:show   | √                      | √                            |
| 更新流日志基本信息 | PUT /v3/{project_id}/enterprise-router/{er_id}/flow-logs/{flow_log_id}         | er:flowlogs:update | √                      | √                            |
| 删除流日志     | DELETE /v3/{project_id}/enterprise-router/{er_id}/flow-logs/{flow_log_id}      | er:flowlogs:delete | √                      | √                            |
| 开启流日志     | POST /v3/{project_id}/enterprise-router/{er_id}/flow-logs/{flow_log_id}/enable | er:flowlogs:enable | √                      | √                            |

| 权限    | 对应API接口                                                                         | 授权项<br>( Action )   | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|-------|---------------------------------------------------------------------------------|---------------------|------------------------|------------------------------|
| 关闭流日志 | POST /v3/{project_id}/enterprise-router/{er_id}/flow-logs/{flow_log_id}/disable | er:flowlogs:disable | √                      | √                            |

## 5.11 标签

| 权限          | 对应API接口                                                         | 授权项<br>( Action ) | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|-------------|-----------------------------------------------------------------|-------------------|------------------------|------------------------------|
| 查询项目标签      | GET /v3/{project_id}/{resource_type}/tags                       | er:tags:list      | √                      | ×                            |
| 查询资源标签      | GET /v3/{project_id}/{resource_type}/{resource_id}/tags         | er:tags:get       | √                      | √                            |
| 创建资源标签      | POST /v3/{project_id}/{resource_type}/{resource_id}/tags        | er:tags:create    | √                      | √                            |
| 批量添加/删除资源标签 | POST /v3/{project_id}/{resource_type}/{resource_id}/tags/action | er:tags:create    | √                      | √                            |
| 删除资源标签      | POST /v3/{project_id}/{resource_type}/{resource_id}/tags/{key}  | er:tags:delete    | √                      | √                            |

## 5.12 配额管理

| 权限   | 对应API接口                                       | 授权项<br>( Action ) | IAM项<br>目<br>(Project) | 企业项目<br>(Enterprise Project) |
|------|-----------------------------------------------|-------------------|------------------------|------------------------------|
| 查询配额 | GET /v3/{project_id}/enterprise-router/quotas | er:quotas:list    | √                      | ×                            |

# A 附录

## A.1 状态码

表 A-1 正常状态码

| 状态码 | 编码         | 说明                                        |
|-----|------------|-------------------------------------------|
| 200 | OK         | GET、PUT、POST操作正常返回。                       |
| 201 | Created    | POST操作正常返回。                               |
| 202 | Accepted   | 常用于需要很长时间才能处理的操作。<br>表示该请求已被接受处理，但处理尚未完成。 |
| 204 | No Content | DELETE操作正常返回。                             |

表 A-2 异常状态码

| 状态码 | 编码                            | 说明                           |
|-----|-------------------------------|------------------------------|
| 400 | Bad Request                   | 服务器未能处理请求。                   |
| 401 | Unauthorized                  | 被请求的页面需要用户名和密码。              |
| 403 | Forbidden                     | 对被请求页面的访问被禁止。                |
| 404 | Not Found                     | 服务器无法找到被请求的页面。               |
| 405 | Method Not Allowed            | 请求中指定的方法不被允许。                |
| 406 | Not Acceptable                | 服务器生成的响应无法被客户端所接受。           |
| 407 | Proxy Authentication Required | 用户必须首先使用代理服务器进行验证，这样请求才会被处理。 |

| 状态码 | 编码                    | 说明                        |
|-----|-----------------------|---------------------------|
| 408 | Request Timeout       | 请求超出了服务器的等待时间。            |
| 409 | Conflict              | 由于冲突，请求无法被完成。             |
| 500 | Internal Server Error | 请求未完成。服务异常。               |
| 501 | Not Implemented       | 请求未完成。服务器不支持所请求的功能。       |
| 502 | Bad Gateway           | 请求未完成。服务器从上游服务器收到一个无效的响应。 |
| 503 | Service Unavailable   | 请求未完成。系统暂时异常。             |
| 504 | Gateway Timeout       | 网关超时。                     |

## A.2 错误码

当您调用API时，如果遇到“APIGW”开头的错误码，请参见[API网关错误码](#)进行处理。

| 状态码 | 错误码         | 错误信息                                                         | 描述                | 处理措施                  |
|-----|-------------|--------------------------------------------------------------|-------------------|-----------------------|
| 400 | ER.04001003 | The enterprise router is unavailable or is being operated.   | 企业路由器实例不可用或正在操作中。 | 请检查企业路由器状态是否可用，并稍后尝试。 |
| 400 | ER.04001104 | The default route table association function is not enabled. | 未使能默认关联开关。        | 请使能默认关联开关后尝试。         |
| 400 | ER.04001105 | The default route table propagation function is not enabled. | 未使能默认传播开关。        | 请使能默认传播开关后尝试。         |
| 400 | ER.04001106 | Invalid ASN.                                                 | ASN参数不合法。         | 请输入合法的ASN参数。          |
| 400 | ER.04002002 | The association already exists.                              | 关联关系已存在。          | 已存在的关联关系，请检查后重试。      |

| 状态码 | 错误码         | 错误信息                                                  | 描述                     | 处理措施                              |
|-----|-------------|-------------------------------------------------------|------------------------|-----------------------------------|
| 400 | ER.04002003 | An operation is being performed on this association.  | 关联正在操作中。               | 请等待关联状态变为可用后重试。                   |
| 400 | ER.04003002 | The propagation already exists.                       | 传播关系已存在。               | 已存在的传播关系，请检查后重试。                  |
| 400 | ER.04003003 | An operation is being performed on this propagation.  | 传播正在操作中。               | 请等待传播状态变为可用后重试。                   |
| 400 | ER.04004002 | This attachment is being used by other resources.     | 连接已被使用，无法删除。           | 请检查是否有其它资源在使用此连接，并在完全清理这些资源后进行重试。 |
| 400 | ER.04004003 | This attachment is unavailable or is being operated.  | 连接不可用或正在操作中。           | 请检查连接状态是否可用，并稍后重试。                |
| 400 | ER.04004004 | This resource is being used by another attachment.    | 资源已被其它连接占用。            | 请检查是否已经为该资源创建了连接。                 |
| 400 | ER.04004101 | This route already exists in the VPC route table.     | 需要创建的路由已经存在 VPC 的路由表中。 | 请关闭配置连接侧路由开关后重试。                  |
| 400 | ER.04004102 | Failed to add the route to the VPC route table.       | 向 VPC 添加固定网段路由失败。      | 请检查 VPC 路由表是否已经存在同网段路由。           |
| 400 | ER.04005003 | This route table is unavailable or is being operated. | 路由表不可用或正在操作中。          | 请检查路由表状态是否可用，并稍后重试。               |
| 400 | ER.04006002 | The destination of the route already exists.          | 路由目的地址已存在。             | 请检查路由配置。                          |

| 状态码 | 错误码         | 错误信息                                                                                         | 描述                                                        | 处理措施                  |
|-----|-------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------------|-----------------------|
| 400 | ER.04006103 | The destination of the route is invalid.                                                     | 路由目的地址不合法。                                                | 请检查目的地址的格式。           |
| 400 | ER.04006104 | The destination cannot be 127.0.0.0/8, 169.254.0.0/16, 224.0.0.0/4, or any of their subnets. | 目的路由地址不能在 127.0.0.0/8, 169.254.0.0/16, 224.0.0.0/4 及其子网内。 | 不能配置该网段路由。            |
| 400 | ER.04006105 | The network instance type of the next hop is not supported.                                  | 路由下一跳的网络实例类型不支持。                                          | 请更换其它类型的网络实例          |
| 400 | ER.04006106 | A blackhole route cannot have a next hop. A non-blackhole route must have a next hop.        | 路由参数非法。                                                   | 请检查路由请求 Body 体。       |
| 400 | ER.04008500 | Flow logs are not supported by the resource type.                                            | 资源类型不支持采集流日志                                              | 请检查开启流日志采集的资源类型       |
| 400 | ER.04008501 | The flow log task does not exist.                                                            | 流日志任务不存在                                                  | 请检查流日志 Id             |
| 400 | ER.04008502 | The flow log task is being operated.                                                         | 流日志任务正在操作中。                                               | 请检查流日志任务状态是否可用，并稍后重试。 |
| 400 | ER.04008503 | The flow log task already exists.                                                            | 流日志采集任务已经存在                                               | 已存在的流日志采集任务，请检查后重试。   |
| 400 | ER.04009005 | Invalid parameters.                                                                          | 无效参数。                                                     | 请检查参数。                |
| 400 | ER.04009008 | Insufficient quota.                                                                          | 资源配额不足。                                                   | 请申请工单扩充配额。            |

| 状态码 | 错误码         | 错误信息                                           | 描述                  | 处理措施       |
|-----|-------------|------------------------------------------------|---------------------|------------|
| 400 | ER.04009009 | The quota type does not exist.                 | 配额类型不存在。            | 请检查查询参数。   |
| 400 | ER.04009011 | The resource is frozen.                        | 资源被冻结无法操作。          | 请将资源解冻后尝试。 |
| 400 | ER.04009012 | This account is frozen or restricted.          | 租户被冻结或受限，无法操作资源。    | 请联系客服处理。   |
| 400 | ER.04009013 | This enterprise router has frozen attachments. | 实例内存在被冻结的连接，无法变更AZ。 | 请联系客服处理。   |
| 400 | ER.04009019 | This feature is not available yet.             | 特性暂未开放。             | 请咨询客服。     |
| 404 | ER.04041001 | This enterprise router does not exist.         | ER实例不存在             | 请检查参数是否正确。 |
| 404 | ER.04042001 | This association does not exist.               | 关联关系不存在。            | 请检查参数是否正确。 |
| 404 | ER.04043001 | This propagation does not exist.               | 传播关系不存在。            | 请检查参数是否正确。 |
| 404 | ER.04044001 | This attachment does not exist.                | 连接不存在。              | 请检查参数是否正确。 |
| 404 | ER.04045001 | This route table does not exist.               | 路由表不存在。             | 请检查参数是否正确。 |
| 404 | ER.04046001 | This route does not exist.                     | 路由不存在。              | 请检查参数是否正确。 |

| 状态码 | 错误码         | 错误信息                                                     | 描述            | 处理措施          |
|-----|-------------|----------------------------------------------------------|---------------|---------------|
| 409 | ER.04091002 | This enterprise router is being used by other resources. | ER实例被使用。      | 请先删除依赖资源后重试。  |
| 409 | ER.04095002 | This route table is being used by other resources.       | 路由表被使用。       | 请先删除依赖资源后重试。  |
| 409 | ER.04095104 | The default association route table cannot be deleted.   | 默认关联路由表无法被删除。 | 请禁用默认路由表关联功能。 |
| 409 | ER.04095105 | The default propagation route table cannot be deleted.   | 默认传播路由表无法被删除。 | 请禁用默认路由表传播功能。 |
| 503 | ER.05039006 | This service is temporarily unavailable.                 | 服务暂时不可用。      | 请联系客服或稍后重试。   |

## A.3 获取项目 ID

### 操作场景

在调用接口的时候，部分URL中需要填入项目ID，所以需要获取到项目ID。有如下两种获取方式：

- [调用API获取项目ID](#)
- [从控制台获取项目ID](#)

### 调用 API 获取项目 ID

项目ID可以通过调用[查询指定条件下的项目列表](#)API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中projects下的“id”即为项目ID。

```
{
  "projects": [
    {
      "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
      "is_domain": false,
    }
  ]
}
```

```
{
  "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
  "name": "project_name",
  "description": "",
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
  },
  "id": "a4adasfjljaaaakla12334jklga9sasfg",
  "enabled": true
},
{
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
}
```

## 从控制台获取项目 ID

从控制台获取项目ID的步骤如下：

1. 登录管理控制台。
2. 鼠标悬停在右上角的用户名，选择下拉列表中的“我的凭证”。  
在“API凭证”页面的项目列表中查看项目ID。

图 A-1 查看项目 ID

