

统一身份认证

API 参考

文档版本 15

发布日期 2017-10-27



版权所有 © 华为技术有限公司 2018。 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,华为公司对本文档内容不做任何明示或默示的声明或保证。

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。

华为技术有限公司

地址: 深圳市龙岗区坂田华为总部办公楼 邮编: 518129

网址: http://www.huawei.com
客户服务邮箱: support@huawei.com

客户服务电话: 4008302118

目 录

1 接口调用方法	
1.1 服务使用方法	1
1.2 请求方法	1
1.3 请求认证方式	2
1.4 Token 认证	2
1.5 AK/SK 认证	3
1.5.1 生成 AK/SK	3
1.5.2 请求签名流程	
1.5.3 示例代码	
1.6 获取项目 ID	12
2 常用消息头	14
2.1 公共请求消息头	14
2.2 公共响应消息头	
3 通过联邦认证机制获取 token	16
4 API 参考	18
4.1.1 查询用户可以访问的租户列表	
4.2 用户	
4.2.1 查询用户列表	
4.2.2 查询用户详情	22
4.2.3 查询用户所属用户组	24
4.2.4 查询用户组所包含的用户	26
4.2.5 创建用户	28
4.2.6 修改密码	31
4.2.7 修改用户信息	33
4.2.8 删除用户	36
4.2.9 删除用户组中用户	37
4.3 用户组	38
4.3.1 查询用户组	38
4.3.2 查询用户组详情	40
4.3.3 创建用户组	41
4.3.4 添加用户到用户组	43

4.3.5 更新用户组	44
4.3.6 删除用户组	
4.3.7 查询用户是否在用户组中	
4.4 项目	
4.4.1 查询指定条件下的项目信息	
4.4.2 查询用户的项目列表	
4.4.3 查询用户可以访问的项目列表	
4.4.4 创建项目	
4.4.5 修改项目	
4.4.6 查询指定项目的信息	57
4.4.7 设置指定项目的状态	59
4.4.8 查询指定项目信息及状态	60
4.5 委托	62
4.5.1 创建委托	62
4.5.2 获取指定委托的详细信息	64
4.5.3 查询指定条件下的委托列表信息	66
4.5.4 修改委托	68
4.5.5 删除委托	70
4.5.6 通过项目给委托赋予权限	71
4.5.7 检查委托在项目上是否有指定权限	73
4.5.8 查询委托在项目上具有的权限列表	74
4.5.9 通过项目删除委托的权限	76
4.5.10 通过租户给委托赋予权限	77
4.5.11 检查委托在租户上是否有指定权限	79
4.5.12 查询委托在租户上具有的权限列表	80
4.5.13 通过租户删除委托的权限	82
4.6 区域	83
4.6.1 查询区域列表	83
4.6.2 查询区域详情	86
4.7 权限	
4.7.1 查询权限列表	
4.7.2 查询权限的详细信息	
4.7.3 查询租户中用户组的权限	
4.7.4 查询项目对应的用户组的权限	
4.7.5 租户的用户组授权	
4.7.6 为项目对应的用户组授权	
4.7.7 删除项目对应的用户组的权限	
4.7.8 删除租户中用户组的权限	
4.7.9 查询租户中用户组是否包含权限	
4.7.10 查询项目对应的用户组是否包含权限	
4.8 Token	
4.8.1 获取用户 Token	

API 参考

4.8.2 通过 assume_role 方式获取用户 token	
4.9 联邦认证	113
4.9.1 身份提供商	113
4.9.1.1 查询身份提供商列表	113
4.9.1.2 查询身份提供商	
4.9.1.3 注册身份提供商	117
4.9.1.4 更新身份提供商	119
4.9.1.5 删除身份提供商	120
4.9.2 映射	
4.9.2.1 查询映射列表	
4.9.2.2 查询映射	123
4.9.2.3 注册映射	126
4.9.2.4 更新映射	
4.9.2.5 删除映射	
4.9.3 协议	
4.9.3.1 查询协议列表	136
4.9.3.2 查询协议	
4.9.3.3 注册协议	139
4.9.3.4 更新协议	141
4.9.3.5 删除协议	142
4.9.4 Metadata	144
4.9.4.1 查询 Metadata 文件	144
4.9.4.2 查询 Keystone 的 Metadata 文件	
4.9.4.3 导入 Metadata 文件	147
4.9.5 token	148
4.9.5.1 获取联邦认证的 unscoped token	148
4.9.5.2 获取联邦认证的 scoped token	150
4.9.6 租户	
4.9.6.1 查询联邦用户可以访问的租户列表	
4.9.7 项目	156
4.9.7.1 查询联邦用户可以访问的项目列表	156
4.9.8 断言	158
4.9.8.1 WebSSO 断言处理接口	
4.9.8.2 ECP 断言处理接口	158
4.10 版本信息	159
4.10.1 查询 Keystone API 的版本信息	159
4.10.2 查询 Keystone API 的 3.0 版本信息	160
4.11 服务和终端节点	161
4.11.1 查询服务列表	161
4.11.2 查询服务详情	164
4.11.3 查询终端节点列表	
4.11.4 查询终端节点详情	167

A 文档修订记录	170
4.12.1 获取用户名称、项目名称、项目 ID	169
4.12 相关参数信息获取	169

1接口调用方法

第三方应用对公有云API的访问需经过签名认证。

本章主要介绍了使用签名的过程和注意事项,并通过示例代码展示了如何使用默认的 Signer对请求进行签名和利用HTTP Client发送请求。

1.1 服务使用方法

公有云API符合REST API的设计理论。

REST从资源的角度来观察整个网络,分布在各处的资源由URI(Uniform Resource Identifier)确定,而客户端的应用通过URL(Unified Resource Locator)来获取资源。

URL的一般格式为(带方括号的为可选项):

https://Endpoint [/uri]

URL中的参数说明,如表1-1所示。

表 1-1 URL 中的参数说明

参数	描述	是否必选
protocol	请求使用的协议类型,例如https。 https表示通过安全的SSL加密的HTTP协议访问该资源。	必选
Endpoint	存放资源的服务器的域名,请从 地区与终端节点 中获取。	必选
uri	资源路径,也即API访问路径。从每个接口的URI段落中获取,例如:获取用户token接口的URI为"v3/auth/tokens"。	可选

1.2 请求方法

在HTTP协议中,可以使用多种请求方法,用于指明以何种方式来访问指定的资源,例如: GET、PUT、POST、DELETE、PATCH。目前提供的REST接口支持的请求方法如下表所示。

表 1-2 请求方法一览表

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。	
PATCH	请求服务器更新资源的部分内容。 当资源不存在的时候,PATCH可能会去创建一个新的资源。	

1.3 请求认证方式

调用接口有如下两种认证方式,您可以任选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。 AK/SK认证安全性更高。

1.4 Token 认证

获取 Token 认证步骤

当您使用Token认证方式时,请采用如下步骤调用接口:

步骤1 发送 "POST https://*IAM的Endpoint*/v3/auth/tokens", 获取IAM的Endpoint及消息体中的区域名称。

请参考地区和终端节点。

步骤2 获取Token,请参考"获取用户Token"。

∭ 说明

- Token值即接口返回消息Header中的"X-Subject-Token"的值。
- Token的有效期为24小时,需要同一个Token鉴权时,可以先缓存起来,避免频繁调用。

步骤3 调用接口时,在业务接口请求消息头中增加"X-Auth-Token",取值为上面步骤中的 Token值。

----结束

获取 Token 认证示例

步骤1 发送 "POST https://API网关地址/v3/auth/tokens",请求内容示例如下:

一说明

下面示例代码中的斜体字需要替换为实际内容,详情请参考"获取用户Token"。

```
{
    "auth": {
```

```
"identity": {
    "methods": [
        "password"
],
    "password": {
        "name": "user_name",
        "password": "password",
        "domain": {
            "name": "domain_name"
        }
    }
},
"scope": {
    "domain": {
        "name": "domain_name"
    }
}
}
```

步骤2 请求响应成功后在响应消息头中包含的"X-Subject-Token"的值即为Token值。

步骤3 调用业务接口,在请求消息头中增加"X-Auth-Token","X-Auth-Token"的取值为前面步骤中获取的Token。

----结束

1.5 AK/SK 认证

通过API网关向下层服务发送请求时,必须使用AK(Access Key ID)、SK(Secret Access Key)对请求进行签名。

□说明

AK: 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:与访问密钥ID结合使用的密钥,对请求进行加密签名,可标识发送方,并防止请求被修改。

1.5.1 生成 AK/SK

步骤1 注册并登录管理控制台,单击右上方登录的账号,选择"我的凭证"。

步骤2 在"我的凭证"页面,选择"管理访问密钥"页签。

步骤3 单击列表下侧的"新增访问密钥",创建新的访问密钥。

每个用户最多可创建2个有效访问密钥(AK/SK),且一旦生成永久有效,为了账户安全性,建议定期更换访问密钥。

步骤4 输入当前用户的登录密码,并通过邮箱或者手机进行验证,输入对应的验证码。

1288

在统一身份认证服务中创建的用户,如果创建时未填写邮箱或者手机号,则只需校验登录密码。

步骤5 单击"确定",下载访问密钥。

∭说明

为防止访问密钥泄露,建议您将其保存到安全的位置。

----结束

1.5.2 请求签名流程

签名前的准备

- 1. 下载API网关签名工具。
 - 下载地址: http://esdk.huawei.com/ilink/esdk/download/HW 456706
- 2. 解压下载的压缩包。
- 3. 创建java工程,将解压出来的jar引用到依赖路径中。

签名过程

步骤1 创建用于签名的请求com.cloud.sdk.DefaultRequest(JAVA)。

步骤2 设置DefaultRequest的目标APIURL、HTTPS方法、内容等信息。

步骤3 对DefaultRequest进行签名:

- 1. 调用SignerFactory.getSigner(String serviceName, String regionName)获取一个签名工具实现的实例。
- 2. 调用Signer.sign(Request<?> request, Credentials credentials)对步骤1创建的请求进行签名。

以下代码展示了这个步骤:

```
//选用签名算法,对请求进行签名
Signer signer = SignerFactory.getSigner(serviceName, region);
//对请求进行签名, request会发生改变
signer.sign(request, new BasicCredentials(this.ak, this.sk));
```

步骤4 把**步骤3**签名产生的request转换为一个适合发送的请求,并将签名后request中的header 信息放入新的request中。

以Apache HttpClient为例,需要把DefaultRequest转换为HttpRequestBase,把签名后的DefaultRequest的header信息放入HttpRequestBase中。

具体过程请查看"1.5.3 示例代码"中的AccessServiceImpl.java。

----结束

1.5.3 示例代码

下面代码展示了如何对一个请求进行签名,并通过HTTP Client发送一个HTTPS请求的过程:

代码分成三个类进行演示:

- AccessService: 抽象类,将GET/POST/PUT/DELETE归一成access方法。
- Demo:运行入口,模拟用户进行GET/POST/PUT/DELETE请求。
- AccessServiceImpl: 实现access方法,具体与API网关通信的代码都在access方法中。

下面示例代码中的region和serviceName(英文服务名缩写),请从**地区和终端节点**中获取。

```
AccessService.java:
package com.cloud.apigateway.sdk.demo;
import java.io.InputStream;
import java.net.URL;
import java.util.Map;
import org.apache.http.HttpResponse;
import com.cloud.sdk.http.HttpMethodName;
public abstract class AccessService {
   protected String serviceName = null;
   protected String region = null;
   protected String ak = null;
   protected String sk = null;
   public AccessService(String serviceName, String region, String ak,
String sk) {
        this.region = region;
       this.serviceName = serviceName;
       this.ak = ak;
        this.sk = sk;
   public abstract HttpResponse access(URL url, Map<String, String>
header, InputStream content, Long contentLength,
       HttpMethodName httpMethod) throws Exception;
   public HttpResponse access(URL url, Map<String, String> header,
HttpMethodName httpMethod) throws Exception {
       return this.access(url, header, null, 01, httpMethod);
   public HttpResponse access(URL url, InputStream content, Long
contentLength, HttpMethodName httpMethod)
       throws Exception {
        return this.access(url, null, content, contentLength,
httpMethod);
   }
   public HttpResponse access (URL url, HttpMethodName httpMethod)
throws Exception {
        return this.access(url, null, null, 01, httpMethod);
   public abstract void close();
   public String getServiceName() {
       return serviceName;
    }
    public void setServiceName(String serviceName) {
       this.serviceName = serviceName;
    }
```

```
public String getRegion() {
    return region;
}

public void setRegion(String region) {
    this.region = region;
}

public String getAk() {
    return ak;
}

public void setAk(String ak) {
    this.ak = ak;
}

public String getSk() {
    return sk;
}

public void setSk(String sk) {
    this.sk = sk;
}
```

AccessServiceImpl.java: package com.cloud.apigateway.sdk.demo; import java.io.IOException; import java.io.InputStream; import java.net.URISyntaxException; import java.net.URL; import java.util.HashMap; import java.util.Map; import javax.net.ssl.SSLContext; import org.apache.http.Header; import org.apache.http.HttpHeaders; import org.apache.http.HttpResponse; import org.apache.http.client.methods.HttpDelete; import org.apache.http.client.methods.HttpGet; import org.apache.http.client.methods.HttpHead; import org.apache.http.client.methods.HttpPatch; import org.apache.http.client.methods.HttpPost; import org.apache.http.client.methods.HttpPut; import org.apache.http.client.methods.HttpRequestBase; import org.apache.http.conn.ssl.AllowAllHostnameVerifier; import org.apache.http.conn.ssl.SSLConnectionSocketFactory; import org.apache.http.conn.ssl.SSLContexts; import org.apache.http.conn.ssl.TrustSelfSignedStrategy; import org.apache.http.entity.InputStreamEntity; import org.apache.http.impl.client.CloseableHttpClient; import org.apache.http.impl.client.HttpClients; import com.cloud.sdk.DefaultRequest; import com.cloud.sdk.Request; import com.cloud.sdk.auth.credentials.BasicCredentials; import com.cloud.sdk.auth.signer.Signer; import com.cloud.sdk.auth.signer.SignerFactory; import com.cloud.sdk.http.HttpMethodName;

```
public class AccessServiceImpl extends AccessService {
private CloseableHttpClient client = null;
public AccessServiceImpl(String serviceName, String region, String ak,
  String sk) {
 super(serviceName, region, ak, sk);
 /** {@inheritDoc} */
 public HttpResponse access(URL url, Map<String, String> headers,
  InputStream content, Long contentLength, HttpMethodName httpMethod)
  throws Exception {
  // Make a request for signing.
 Request request = new DefaultRequest(this.serviceName);
  try {
  \ensuremath{//} Set the request address.
  request.setEndpoint(url.toURI());
  String urlString = url.toString();
  String parameters = null;
    if (urlString.contains("?")) {
        parameters = urlString.substring(urlString.indexOf("?") + 1);
        Map parametersmap = new HashMap<String, String>();
        if (null != parameters && !"".equals(parameters)) {
                    String[] parameterarray = parameters.split("&");
                    for (String p : parameterarray) {
                        String key = p.split("=")[0];
                        String value = p.split("=")[1];
                        parametersmap.put(key, value);
                    }
                    request.setParameters(parametersmap);
          }
     }
  } catch (URISyntaxException e) {
  // It is recommended to add logs in this place.
  e.printStackTrace();
  // Set the request method.
 request.setHttpMethod(httpMethod);
 if (headers != null) {
  // Add request header information if required.
  request.setHeaders(headers);
  // Configure the request content.
 request.setContent(content);
  // Select an algorithm for request signing.
 Signer signer = SignerFactory.getSigner(serviceName, region);
  // Sign the request, and the request will change after the signing.
  signer.sign(request, new BasicCredentials(this.ak, this.sk));
// Make a request that can be sent by the HTTP client.
```

```
HttpRequestBase httpRequestBase = createRequest(url, null,
   request.getContent(), contentLength, httpMethod);
 Map<String, String> requestHeaders = request.getHeaders();
  // Put the header of the signed request to the new request.
 for (String key : requestHeaders.keySet()) {
  if (key.equalsIgnoreCase(HttpHeaders.CONTENT LENGTH.toString())) {
   continue;
  httpRequestBase.addHeader(key, requestHeaders.get(key));
 HttpResponse response = null;
 SSLContext sslContext = SSLContexts.custom()
    .loadTrustMaterial(null, new TrustSelfSignedStrategy())
    .useTLS().build();
 SSLConnectionSocketFactory sslSocketFactory = new
SSLConnectionSocketFactory(
    sslContext, new AllowAllHostnameVerifier());
 client = HttpClients.custom().setSSLSocketFactory(sslSocketFactory)
    .build();
 // Send the request, and a response will be returned.
 response = client.execute(httpRequestBase);
 return response;
 * Make a request that can be sent by the HTTP client.
  * @param url
              specifies the API access path.
  * @param header
               specifies the header information to be added.
  * @param content
              specifies the body content to be sent in the API call.
  * @param contentLength
              specifies the length of the content. This parameter is
optional.
 * @param httpMethod
              specifies the HTTP method to be used.
  * @return specifies the request that can be sent by an HTTP client.
private static HttpRequestBase createRequest(URL url, Header header,
  InputStream content, Long contentLength, HttpMethodName httpMethod) {
 HttpRequestBase httpRequest;
 if (httpMethod == HttpMethodName.POST) {
  HttpPost postMethod = new HttpPost(url.toString());
  if (content != null) {
   InputStreamEntity entity = new InputStreamEntity(content,
     contentLength);
   postMethod.setEntity(entity);
  httpRequest = postMethod;
  } else if (httpMethod == HttpMethodName.PUT) {
  HttpPut putMethod = new HttpPut(url.toString());
  httpRequest = putMethod;
  if (content != null) {
   InputStreamEntity entity = new InputStreamEntity(content,
     contentLength);
```

```
putMethod.setEntity(entity);
  } else if (httpMethod == HttpMethodName.PATCH) {
  HttpPatch patchMethod = new HttpPatch(url.toString());
  httpRequest = patchMethod;
  if (content != null) {
   InputStreamEntity entity = new InputStreamEntity(content,
     contentLength);
   patchMethod.setEntity(entity);
  } else if (httpMethod == HttpMethodName.GET) {
  httpRequest = new HttpGet(url.toString());
  } else if (httpMethod == HttpMethodName.DELETE) {
  httpRequest = new HttpDelete(url.toString());
  } else if (httpMethod == HttpMethodName.HEAD) {
  httpRequest = new HttpHead(url.toString());
  } else {
  throw new RuntimeException ("Unknown HTTP method name: "
    + httpMethod);
 httpRequest.addHeader(header);
 return httpRequest;
 @Override
 public void close() {
 try {
  if (client != null) {
   client.close();
  } catch (IOException e) {
  // It is recommended to add logs in this place.
  e.printStackTrace();
}
Demo.java:
package com.cloud.apigateway.sdk.demo;
import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import org.apache.http.HttpResponse;
import com.cloud.sdk.http.HttpMethodName;
public class Demo {
 //replace real region
private static final String region = "regionName";
 //replace real service name
private static final String serviceName = "serviceName";
```

```
public static void main(String[] args) {
 //replace real AK
 String ak = "akString";
 //replace real SK
 String sk = "skString";
 // get method
 //replace real url
 String url = "urlString";
 get(ak, sk, url);
 // post method
 //replace real url
 String postUrl = "urlString";
 //replace real body
 String postbody = "bodyString";
 post(ak, sk, postUrl, postbody);
 // put method
 //replace real body
 String putbody = "bodyString";
 //replace real url
 String putUrl = "urlString";
 put(ak, sk, putUrl, putbody);
 // delete method
 //replace real url
 String deleteUrl = "urlString";
 delete(ak, sk, deleteUrl);
public static void put (String ak, String sk, String requestUrl,
  String putBody) {
 AccessService accessService = null;
  accessService = new AccessServiceImpl(serviceName, region, ak, sk);
  URL url = new URL(requestUrl);
  HttpMethodName httpMethod = HttpMethodName.PUT;
  InputStream content = new ByteArrayInputStream(putBody.getBytes());
  HttpResponse response = accessService.access(url, content,
    (long) putBody.getBytes().length, httpMethod);
  System.out.println(response.getStatusLine().getStatusCode());
 } catch (Exception e) {
  e.printStackTrace();
 } finally {
  accessService.close();
 }
public static void patch (String ak, String sk, String requestUrl,
  String putBody) {
 AccessService accessService = null;
trv {
```

```
accessService = new AccessServiceImpl(serviceName, region, ak, sk);
 URL url = new URL(requestUrl);
 HttpMethodName httpMethod = HttpMethodName.PATCH;
 InputStream content = new ByteArrayInputStream(putBody.getBytes());
 HttpResponse response = accessService.access(url, content,
    (long) putBody.getBytes().length, httpMethod);
 System.out.println(convertStreamToString(response.getEntity()
   .getContent());
 } catch (Exception e) {
 e.printStackTrace();
 } finally {
 accessService.close();
}
public static void delete(String ak, String sk, String requestUrl) {
AccessService accessService = null;
try {
 accessService = new AccessServiceImpl(serviceName, region, ak, sk);
 URL url = new URL(requestUrl);
 HttpMethodName httpMethod = HttpMethodName.DELETE;
 HttpResponse response = accessService.access(url, httpMethod);
 System.out.println(convertStreamToString(response.getEntity()
   .getContent());
 } catch (Exception e) {
 e.printStackTrace();
 } finally {
 accessService.close();
public static void get(String ak, String sk, String requestUrl) {
AccessService accessService = null;
t.rv {
 accessService = new AccessServiceImpl(serviceName, region, ak, sk);
 URL url = new URL(requestUrl);
 HttpMethodName httpMethod = HttpMethodName.GET;
 HttpResponse response;
 response = accessService.access(url, httpMethod);
 System.out.println(convertStreamToString(response.getEntity()
   .getContent());
 } catch (Exception e) {
 e.printStackTrace();
} finally {
 accessService.close();
}
public static void post(String ak, String sk, String requestUrl,
 String postbody) {
AccessService accessService = new AccessServiceImpl(serviceName,
region, ak, sk);
```

```
URL url = null;
try {
 url = new URL(requestUrl);
 } catch (MalformedURLException e) {
 e.printStackTrace();
InputStream content = new ByteArrayInputStream(postbody.getBytes());
HttpMethodName httpMethod = HttpMethodName.POST;
HttpResponse response;
 response = accessService.access(url, content,
    (long) postbody.getBytes().length, httpMethod);
 System.out.println(convertStreamToString(response.getEntity()
   .getContent()));
 } catch (Exception e) {
 e.printStackTrace();
 } finally {
 accessService.close();
private static String convertStreamToString(InputStream is) {
BufferedReader reader = new BufferedReader(new InputStreamReader(is));
StringBuilder sb = new StringBuilder();
String line = null;
try {
 while ((line = reader.readLine()) != null) {
  sb.append(line + "\n");
} catch (IOException e) {
 e.printStackTrace();
} finally {
 try {
  is.close();
 } catch (IOException e) {
  e.printStackTrace();
return sb.toString();
```

□ 说明

- 1. URI、AK、SK、HTTP METHOD是必须设置的参数。
- 2. 可通过request.addHeader()添加头信息。

1.6 获取项目 ID

在调用接口的时候,部分URL中需要填入项目ID,所以需要在"我的凭证"页面获取项目ID。

项目ID获取步骤如下:

步骤1 注册并登录管理控制台。

步骤2 单击登录账户并选择"我的凭证"。

步骤3 在"我的凭证"页面的"项目列表"中查看项目ID。

----结束

2 常用消息头

REST公共消息头包含两类:公共请求消息头和公共响应消息头,本章对这部分内容进行介绍。

2.1 公共请求消息头

表 2-1 公共请求消息头

参数	描述	是否必选	示例
x-sdk-date	请求的发生时间,格式为 (YYYYMMDD'T'HHMM SS'Z')。 取值为当前系统的GMT 时间。	否 使用AK/SK认 证时必选。	20150907T101459Z
Authorization	签名认证信息。 该值来源于请求签名结 果。 请参考 请求签名流程 。	否 使用AK/SK认 证时必选。	SDK-HMAC-SHA256 Credential=ZIRRKMT WPTQFQI1WKNKB/ 20150907//ec2/ sdk_request, SignedHeaders=content- type;host;x-sdk-date, Signature=55741b610f3 c9fa3ae40b5a8021ebf7e bc2a28a603fc62d25cb3b fe6608e1994
Host	请求的服务器信息,从服务API的URL中获取。值为hostname[:port]。未写端口时使用默认的端口,https的默认端口为443。	否 使用AK/SK认 证时必选。	code.test.com 或者 code.test.com:443
Content-type	发送的实体的MIME类型。	是	application/json

参数	描述	是否必选	示例
Content- Length	请求消息体长度,单位为 Byte。	POST/PUT请 求必填。GET 不能包含。	3495
X-Project-Id	project id,用于获取不同 project的token。	否	e9993fc787d94b6c886cb aa340f9c0f4
X-Auth-Token	用户token。	否 使用token认证 时必选。	-

∭说明

其它header属性,请遵照http协议。

2.2 公共响应消息头

表 2-2 公共响应消息头

参数	描述
Content-Length	响应消息体的字节长度,单位为Byte。
Date	系统响应的时间。
Content-type	发送的实体的MIME类型。

3 通过联邦认证机制获取 token

统一命令行客户端工具的安装需要使用root权限,以下配置Openstack Client的操作只需要普通用户权限。



注意

接口调用操作应该在一个安全的网络环境中进行(在VPN或者在租户的云服务器中),如果在不安全的网络环境中,可能会收到中间人攻击。

步骤1 使用文本编辑器创建环境变量文件,在文件中设置用户名、密码、区域、SAML协议版本、IAM地址和端口等信息。参数说明如表4-1所示。

示例如下:

export OS IDENTITY API VERSION=3

export OS_AUTH_TYPE=v3samlpassword

export OS_AUTH_URL=https://iam.cn-north-1.myhwclouds.com:443/v3

export OS IDENTITY PROVIDER=idpid

export OS_PROTOCOL=saml

export OS_IDENTITY_PROVIDER_URL=https://idp.example.com/idp/profile/SAML2/SOAP/ECP

export OS_USERNAME=username

export OS_PASSWORD=userpassword

export OS DOMAIN NAME=example-domain-name

表 3-1 环境变量文件参数说明

参数名称	说明	
OS_IDENTITY_API_VERSION	认证接口版本,固定值为3。	
OS_AUTH_TYPE	认证类型,固定值为v3samlpassword。	

参数名称	说明
OS_AUTH_URL	格式为"https:// <i>IAM地址:端口号/接口版本</i> "。 ● 端口号: 固定值为443。 ● 接口版本: 固定值为v3。
OS_IDENTITY_PROVIDER	身份提供商唯一ID,注册Identity Provider时指定的ID。
OS_DOMAIN_NAME	待认证的租户名称。
OS_PROTOCOL	SAML协议版本,固定值为saml。
OS_IDENTITY_PROVIDER_URL	Identity Provider处理通过客户端代理机制 (ECP)发起的认证请求的地址。
OS_USERNAME	用户在Identity Provide认证时使用的用户名。
OS_PASSWORD	用户在Identity Provide认证时使用的密码。

步骤2 执行如下命令,设置环境变量。

source keystonerc

步骤3 执行如下命令,获取token。

openstack token issue

----结束

4 API 参考

IAM通过如下安全协议和行为来保证API接口的安全性:

- 1. 所有API接口都使用HTTPS协议。
- 2. 首先,客户端使用正确的凭证从服务端获取token;然后,客户端在调用服务端API的同时会将token发送给服务端;最后,服务端通过校验客户端token的有效性来决定是否为客户端提供服务。
- 3. API接口提供鉴权能力,因此拥有相应权限的的客户端才能成功调用对应的API。

4.1 租户

4.1.1 查询用户可以访问的租户列表

功能介绍

该接口用于查询用户可以访问的租户列表。

URI

URI格式

GET /v3/auth/domains

请求

● Request Header参数说明

参数	是否必选	类型	说明
X-Auth-Token	是	String	已认证的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/auth/domains

响应

● Response Body参数说明

参数	是否必 选	类型	说明
domain s	是	JSONArray	租户列表。
links	是	JSONObject	资源访问链接。

● domain格式说明

参数	是否必选	类型	说明
enabled	是	Boolean	是否启用租户,true为启用,false为停 用,默认为true。
id	是	String	租户ID。
enterpriseN ame	是	String	企业租户名。
name	是	String	租户名。
links	是	JSONObj ect	租户资源的链接。
description	否	String	租户描述。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。

状态码	说明
403	鉴权失败。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2 用户

4.2.1 查询用户列表

功能介绍

该接口用于查询用户列表。

URI

- URI格式 GET /v3/users
- 参数说明

参数	是否必选	类型	说明
domain_id	否	String	用户所属的 domain ID。
enabled	否	String	是否启用用户, true为启用,false 为停用,默认为 true。
name	否	String	用户名。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。

参数	是否必选	类型	说明
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限 的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/users

响应

● Response Body参数说明

参数	是否必选	类型	说明
users	是	JSONArray	user对象数组。
links	是	JSONObject	资源访问链接。

● user格式说明

参数	是否必选	类型	说明
enabled	是	Boolean	是否启用用户, true为启用,false 为停用,默认为 true。
id	是	String	用户ID。
domain_id	是	String	用户所在domain 的id。
name	是	String	用户名。
links	是	JSONObject	用户资源的链 接。
default_project_id	否	String	用户默认的project ID。
password_expires_ at	是	Strng	密码过期时间 (UTC时间), null表示密码不过 期。

● 响应样例

```
{
    "users": [{
        "name": "james1234",
        "links": {
              "self": "https://sample.domain.com/v3/users/6d8b04e3bf99445b8f76300903e5bf32"
        },
        "description": "james1234",
```

```
"domain_id": "88b16b6440684467b8825d7d96e154d8",
    "enabled": false,
    "id": "6d8b04e3bf99445b8f76300903e5bf32",
    "default_project_id": "263fd9",
    "password_expires_at": "2016-12-07T00:00:00.000000Z"
}],
    "links": {
        "self": "https://sample.domain.com/v3/users?
domain_id=88b16b6440684467b8825d7d96e154d8&enabled=false",
        "previous": null,
        "next": null
}
```

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.2 查询用户详情

功能介绍

该接口用于查询指定用户的详细信息。

URI

- URI格式 GET /v3/users/{user_id}
- 参数说明

参数	是否必选	类型	说明
user_id	是	String	用户ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限的 token,或用户自身 的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/users/43cbe5e77aaf4665bbb962062dc1fc9d

响应

● Response Body参数说明

参数	是否必选	类型	说明
user	是	JSONObject	user对象。

● user格式说明

参数	是否必选	类型	说明
enabled	是	Boolean	是否启用用户。
id	是	String	用户ID。
domain_id	是	String	用户所在domain 的id。
name	是	String	用户名称。
links	是	JSONObject	用户资源的链 接。
default_project_id	否	String	用户默认的project ID。
password_expires_ at	是	String	密码过期时间 (UTC时间), null表示密码不过 期。
description	否	String	用户描述。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.3 查询用户所属用户组

功能介绍

该接口用于查询指定用户所属的用户组信息。

URI

● URI格式

GET /v3/users/{user_id}/groups

● 参数说明

参数	是否必选	类型	说明
user_id	是	String	用户ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限 的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/users/
43cbe5e77aaf4665bbb962062dc1fc9d/groups

响应

● Response Body参数说明

参数	是否必选	类型	说明
groups	是	JSONArray	用户组列表。
links	是	JSONObject	资源访问链接。

● group格式说明

参数	是否必选	类型	说明
description	是	String	用户组描述。
id	是	String	用户组ID。
domain_id	是	String	用户组所在domain的 id。
name	是	String	用户组名称。
links	是	JSONObject	用户组资源访问链接。

● 响应样例

}

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.4 查询用户组所包含的用户

功能介绍

该接口提供查询用户组中所包含的用户的功能。

URI

● URI格式 GET /v3/groups/{group_id}/users

● URI参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。

● 查询参数说明

属性	是否必选	类型	说明
domain_id	否	String	用户组所属Domain的ID。
name	否	String	用户的名称,长度小于等于 64 字 节。
enabled	否	String	用户是否启用,true是启用,false 是停用。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth、 op_service权限或当前租户下具有 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/groups/
00007111583e457389b0d4252643181b/users

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
links	是	JSONObject	用户组内用户的链接,包含 next,previous,self。
users	是	JSONArray	用户组内用户的列表。

● group格式说明

参数	是否必选	类型	说明
description	是	String	用户的描述。
domain_id	是	String	用户所属Domain的ID。
enabled	是	Boolean	用户是否启用的标志位。
id	是	String	用户的ID。
links	是	JSONObject	用户的链接。
name	是	String	用户的名称。

● 响应样例

```
"previous": null,
    "next": null
}
```

状态码

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.2.5 创建用户

功能介绍

该接口用于在某一租户下创建用户。

URI

URI格式

POST /v3/users

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限的 token。

● Request Body参数说明

参数	是否必选	类型	说明
name	是	String	用户名,长度5~32之间,只 能包含大写字母、小写字 母、空格、数字、特殊字符 ()且不能以数字开头。
domain_id	否	String	用户所在domain的ID。
enabled	否	Boolean	是否启用用户,true为启用,false为停用,默认为true。
password	否	String	用户密码。 长度在6~32之间; 至少包含以下四种字符中的两种:大写字母、小写字母、外写字母、数字和特殊字符; 必须满足账户设置中密码策略的要求; 密码最小长度支持用户定制,参数"password"支持的最小密码长度以用户定制的策略为基准。
default_project_id	否	String	用户默认的project ID。

● 请求样例

```
1. 按照以下模板创建临时文件${filename}.json。${filename}是临时文件名,用户可以自定义。
{
        "user": {
             "default_project_id": "acf2ffabba974fae8f30378ffde2cfa6",
             "domain_id": "88b16b6440684467b8825d7d96e154d8",
             "enabled": true,
             "name": "jamesdoe",
             "password": "********"
        }
}
2. 在${filename}.json文件同目录下,执行如下curl命令。
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H
        "%-Auth-Token:$token" -X POST -d @${filename}.json https://172.30.48.86:31943/v3/users
3. 在${filename}.json文件同目录下,执行如下命令删除临时文件${filename}.json。

rm ${filename}.json
```

响应

● Response Body参数说明

参数	是否必选	类型	说明
user	是	JSONObject	user对象。

● user格式说明

参数	是否必选	类型	说明
enabled	是	Boolean	是否启用用户,true为启 用,false为停用,默认 为true。
id	是	String	用户ID。
domain_id	是	String	用户所在domain的id。
name	是	String	用户名称。
links	是	JSONObject	用户资源的链接。
default_project_id	否	String	用户默认的project ID。
password_expires_ at	是	String	密码过期时间(UTC时间),null表示密码不过期。

● 响应样例

状态码

状态码	说明
201	创建成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
409	资源冲突。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.6 修改密码

功能介绍

该接口用于用户修改自己密码。

URI

● URI格式

POST /v3/users/{user_id}/password

● URI参数说明

参数	是否必选	类型	说明
user_id	是	String	用户ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8" 。
X-Auth-Token	是	String	用户自身的token。

● Request Body参数说明

参数	是否必选	类型	说明
original_password	是	String	用户的原密码。

参数	是否必选	类型	说明
password	是	String	用户的新密码。 • 长度6~32位字符; • 至少包含以下四种字符中的两种:大写字母、小写字母、数字和特殊字符;
			● 不能包含手机号和邮箱; ● 必须满足账户设置中密码策略的要求; ● 密码最小长度支持用户定制,参数"password"支持的最小密码长度以用户定制的策略为基准。

● 请求样例

响应

无响应体。

状态码	说明
204	修改成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。

状态码	说明
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.7 修改用户信息

功能介绍

该接口用于修改租户下对应的用户信息。

URI

- URI格式 PATCH /v3/users/{user_id}
- URI参数说明

参数	是否必选	类型	说明
user_id	是	String	用户ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限 的token。

● Request Body参数说明

参数	是否必选	类型	说明
name	否	String	用户名,长度5~32之间,只能包含大写字母、小写字母、空格、数字、特殊字符()且不能以数字开头。
domain_id	否	String	用户所在domain的ID。

参数	是否必选	类型	说明
enabled	否	Boolean	是否启用用户,true为启用,false为停用,默认为true。
password	否	String	修改后的用户密码。 长度在6~32之间; 至少包含以下四种字符中的两种:大写字母、小写字母、小写字母、数字和特殊字符; 不能包含手机号和邮箱; 必须满足账户设置中密码策略的要求; 密码最小长度支持用户定制,参数"password"支持的最小密码长度以用户定制的策略为基准。
default_project_id	否	String	用户默认的project ID。

● 请求样例

响应

● Response Body参数说明

参数	是否必选	类型	说明
user	是	JSONObject	user对象。

● user格式说明

参数	是否必选	类型	说明
enabled	是	Boolean	是否启用用户,ture为启 用,false为停用,默认为 true。
id	是	String	用户ID。

参数	是否必选	类型	说明
domain_id	是	String	用户所在domain的id。
name	是	String	用户名称。
links	是	JSONObj ect	用户资源的链接。
default_project_id	否	String	用户默认的project ID。
password_expires_a t	是	String	密码过期时间(UTC时 间),null表示密码不过 期。

● 响应样例

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
409	资源冲突。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.8 删除用户

功能介绍

该接口用于删除指定用户。

URI

● URI格式

DELETE /v3/users/{user_id}

● URI参数说明

参数	是否必选	类型	说明
user_id	是	String	用户ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限 的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X DELETE https://172.30.48.86:31943/v3/users/
2c1c6c54e59141b889c99e6fada5f19f

响应

无响应体。

状态码	说明
204	删除成功。
400	请求错误。
401	认证失败。
403	鉴权失败。

状态码	说明
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.2.9 删除用户组中用户

功能介绍

该接口提供删除用户组中用户的功能。

URI

- URI格式
 - DELETE /v3/groups/{group_id}/users/{user_id}
- URI参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。
user_id	是	String	用户的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限、当前租户下具有secu_admin或 op_service权限的用户token)。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X DELETE https://172.30.48.86:31943/v3/groups/
00007111583e457389b0d4252643181b/users/edb66d2b656c43d0b67fb143d670bb3a

响应

无响应体。

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.3 用户组

4.3.1 查询用户组

功能介绍

该接口提供查询用户组信息的功能。

URI

● URI格式 GET /v3/groups {?domain_id,name}

● 查询参数说明

属性	是否必选	类型	说明
domain_id	否	String	用户组所在Domain的ID。
name	否	String	用户组的名称,长度小于等于64 字节。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(当前租户下具有 secu_admin或op_service权限的用 户token)。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json; charset=utf8' -X GET https://10.145.93.56:31943/v3/groups?domain_id=ac7197fd67a24dc5850972854729a762&name=group123

响应

● Reponse Body参数说明

参数	是否必 选	类型	说明
links	是	JSONObject	用户组的链接,包含 next,previous,self。
groups	是	JSONArray	用户组列表。

● group参数说明

参数	是否必 选	类型	说明
description	是	String	用户组的描述。
domain_id	是	String	用户组所属Domain的ID。
id	是	String	用户组的ID。
links	是	JSONObject	用户组的链接。
name	是	String	用户组的名称。
create_time	是	Long	用户组创建时间。

● 响应样例

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。

状态码	说明
403	没有操作权限。

4.3.2 查询用户组详情

功能介绍

该接口提供查询用户组详细信息的功能。

URI

● URI格式

GET /v3/groups/{group_id}

● 查询参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(当前租户下具有 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://10.145.93.56:31943/v3/groups/ab9f261180d746ef8624beb5ae39b5aa

响应

● Reponse Body参数说明

参数	是否必 选	类型	说明
group	是	JSONObject	用户组的响应体。
description	是	String	用户组的描述。
domain_id	是	String	用户组所属Domain的ID。
id	是	String	用户组的ID。

参数	是否必 选	类型	说明
links	是	JSONObject	用户组的链接。
name	是	String	用户组的名称。
create_time	是	Long	用户组创建时间。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.3.3 创建用户组

功能介绍

该接口提供创建用户组的功能。

URI

URI格式

POST /v3/groups

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限或当前租户下具有secu_admin权限的用户token)。

● Request Body参数说明

参数	是否必选	类型	说明
group	是	JSONObject	用户组的请求体。
description	是	String	用户组的描述,长度小于等于255 字节。
domain_id	否	String	用户组所属Domain的ID。
name	是	String	用户组的名称,长度小于等于64 字节。

● 请求样例

curl -i - k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token: \$token" -X POST -d' ("group": {"description": "Contract developers", "domain_id": "d54061ebcb5145dd814f8eb3fe9b7ac0", "name": "jixiang2"}}' https://172.30.48.86:31943/v3/groups

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
group	是	JSONObject	用户组的响应体。
description	是	String	用户组的描述。
domain_id	是	String	用户组所属Domain的ID。
id	是	String	用户组的ID。
links	是	JSONObject	用户组的链接。
name	是	String	用户组的名称。

● 响应样例

状态码	说明
201	创建成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
409	资源冲突。

4.3.4 添加用户到用户组

功能介绍

该接口提供为用户组添加用户的功能。

URI

- URI格式
 - PUT /v3/groups/{group_id}/users/{user_id}
- URI参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。
user_id	是	String	用户的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限、当前租户下具有secu_admin或 op_service权限的用户token)。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X PUT https://172.30.48.86:31943/v3/groups/
00007111583e457389b0d4252643181b/users/edb66d2b656c43d0b67fb143d670bb3a

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.3.5 更新用户组

功能介绍

该接口提供更新用户组信息的功能。

URI

- URI格式 PATCH /v3/groups/{group_id}
- URI参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(当前租户下具有 secu_admin权限的用户token)。

● Request Body参数说明

参数	是否必选	类型	说明
group	是	Object	group请求体。

参数	是否必选	类型	说明
description	否	String	用户组的描述,长度小于等于255 字节。
domain_id	否	String	用户组所属Domain的ID。
name	否	String	用户组的名称,长度小于等于64 字节。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X PATCH -d' {"group": {"description": "Contract developers 2016"}}' https://172.30.48.86:31943/v3/groups/aaec2abd4eba430fbf61541ffde76650

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
group	是	Dict	用户组的响应体。
description	是	String	用户组描述。
domain_id	是	String	用户组所属Domain的ID。
id	是	String	用户组的ID。
links	是	Dict	用户组的链接。
name	是	String	用户组的名称。

● 响应样例

```
{
    "group": {
        "domain_id": "d54061ebcb5145dd814f8eb3fe9b7ac0",
        "description": "Contract developers 2016",
        "id": "aaec2abd4eba430fbf61541ffde76650",
        "links": {
            "self": "https://sample.domain.com/v3/groups/aaec2abd4eba430fbf61541ffde76650"
        },
        "name": "jixiang1"
    }
}
```

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。

状态码	说明
404	找不到资源。
409	资源冲突。
501	接口没有实现。

4.3.6 删除用户组

功能介绍

该接口提供删除用户组的功能。

URI

- URI格式
 DELETE /v3/groups/{group_id}
- URI参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限或当前租户下具有secu_admin权限的用户token)。

● 请求程例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X DELETE https://172.30.48.86:31943/v3/groups/aaec2abd4eba430fbf61541ffde76650

响应

无响应体。

状态码	说明
204	删除成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.3.7 查询用户是否在用户组中

功能介绍

该接口查询某用户是否属于某用户组的功能。

URI

- URI格式 HEAD /v3/groups/{group_id}/users/{user_id}
- URI参数说明

属性	是否必选	类型	说明
group_id	是	String	用户组的ID。
user_id	是	String	用户的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限或当前租户下具有secu_admin权限的用户token)。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X HEAD https://172.30.48.86:31943/v3/groups/
00007111583e457389b0d4252643181b/users/edb66d2b656c43d0b67fb143d670bb3a

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.4 项目



注意

不允许在MOS项目下创建任何云服务资源,否则可能会导致资源无法管理。

4.4.1 查询指定条件下的项目信息

功能介绍

该接口用于查询指定条件下的项目信息。

URI

● URI格式

GET /v3/projects {?domain_id,name,enabled,parent_id,is_domain,page,per_page}

● 参数说明

参数	是否必选	类型	说明	
domain_id	否	String	用户所属企业账户的id。	
name	否	String	project的名称。	
parent_id	否	String	项目继承的父项目ID。	
enabled	否	Boolean	项目是否启用。	
is_domain	否	Boolean	是否为租户。	
page	否	Integer	查询第几页的数据,查询值最小 为1。	

参数	是否必选	类型	说明
per_page	否	Integer	每页的数据个数,取值范围为 [1,5000]。

∭说明

- 该接口至少需要一种查询条件。禁止全量查询project列表。
- 需要分页查询时,必须保证查询参数中同时存在page和per_page。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	String	已认证的具有op_service权限,或 需要查询的目标租户的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -X "X-Auth-Token:\$token" -X GET https://10.145.93.56:31943/v3/projects?
domain_id=5c9f5525d9d24c5bbf91e74d86772029&name=region_name

响应

● Response Body参数说明

参数	是否必选	类型	说明
projects	是	List	project列表。

● projects格式说明

参数	是否必选	类型	说明	
description	是	String	project描述。	
enabled	是	Boolean	project是否可用。	
id	是	String	project id.	
domain_id	是	String	project所在domain的id。	
name	是	String	project名称。	
is_domain	是	Boolean	是否为domain	
enabled	是	Boolean	project是否可用	
parent_id	是	String	project的父ID	

● 响应样例

```
{
    "links": {
        "self": "www.example.com/v3/projects?
domain_id=c9f5525d9d24c5bbf9le74d86772029&name=region_name",
        "previous": null,
        "next": null
    },
    "projects": [
        {
             "is_domain": false,
            "description": "",
            "links": {
                  "self": "www.example.com/v3/projects/e86737682ab64b2490c48f08bcc41914"
        },
            "enabled": true,
            "id": "e86737682ab64b2490c48f08bcc41914",
            "parent_id": "c9f5525d9d24c5bbf9le74d86772029",
            "domain_id": "c9f5525d9d24c5bbf9le74d86772029",
            "name": "region_name"
        }
    }
}
```

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
500	内部服务错误。
503	服务不可用。

4.4.2 查询用户的项目列表

功能介绍

该接口用于查询指定用户可访问的项目信息。

URI

● URI格式

GET /v3/users/{user_id}/projects

● 参数说明

参数	是否必选	类型	说明
user_id	是	String	用户ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的拥有 Security Administrator权限 的token,或用户 自身的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/users/
43cbe5e77aaf4665bbb962062dc1fc9d/projects

响应

● Response Body参数说明

参数	是否必选	类型	说明
projects	是	Array	project列表。
links	是	Object	资源访问链接。

● project格式说明

参数	是否必选	类型	说明
description	是	String	project描述。
id	是	String	project ID.
domain_id	是	String	project所在domain的id。
name	是	String	project名称。
links	是	Object	project资源访问链接。
is_domain	是	Boolean	是否为domain
enabled	是	Boolean	project是否可用
parent_id	是	String	project的父ID

● 响应样例

```
{
    "links": {
        "self": "www.example.com/v3/auth/projects",
        "previous": null,
        "next": null
```

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.4.3 查询用户可以访问的项目列表

功能介绍

该接口用于查询用户可以访问的项目列表。

URI

URI格式

GET /v3/auth/projects

请求

● Request Header参数说明

参数	是否必选	类型	说明
X-Auth-Token	是	String	已认证的需要查 询的用户的 token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET /v3/auth/projects

响应

● Response Body参数说明

参数	是否必选	类型	说明
projects	是	array	项目列表。
links	是	Object	资源访问链接。

● project格式说明

参数	是否必选	类型	说明
description	是	String	project描述。
id	是	String	project ID.
domain_id	是	String	project所在domain的id。
name	是	String	project名称。
links	是	Object	project资源访问链接。
is_domain	是	Boolean	是否为domain
enabled	是	Boolean	project是否可用
parent_id	是	String	project的父ID

● 响应样例

```
{
  "links": {
      "self": "www.example.com/v3/auth/projects",
      "previous": null,
      "next": null
},
    "projects": [
      {
            "is_domain": false,
            "description": "",
            "links": {
```

```
"self": "www.example.com/v3/projects/9041929bcc6e4bfe85add4e7b96ffdd7"
},

"enabled": true,
"id": "9041929bcc6e4bfe85add4e7b96ffdd7",

"parent_id": "398998b5392f4150ad48fe456d6de4f1",

"domain_id": "398998b5392f4150ad48fe456d6de4f1",

"name": "region"
},

{

"is_domain": false,
"description": "",
"links": {

"self": "www.example.com/v3/projects/ee65ca70d3cf43aaalea6492ce15f289"
},

"enabled": true,
"id": "ee65ca70d3cf43aaalea6492ce15f289",

"parent_id": "398998b5392f4150ad48fe456d6de4f1",
"domain_id": "398998b5392f4150ad48fe456d6de4f1",
"name": "MOS"
}

]
```

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.4.4 创建项目

功能介绍

该接口提供创建项目功能。

URI

URI格式

POST /v3/projects

请求

● Request Header参数说明

参数	是否为必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	String	有效的token(当前租户下具有 secu_admin的用户token)。

● Request Body参数说明

属性	是否为必选	类型	说明
name	是	String	项目名,必须以存在的区域ID_开 头,长度小于等于64。 示例: <i>{region_id}_</i> test1。
parent_id	否	String	项目所属的父项目ID。
domain_id	否	String	项目所在租户的ID。
description	否	String	项目描述,长度小于等于255。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X POST -d '{"project":{"domain_id":"acf2ffabba974fae8f30378ffde2cfa6", "name":"region_test1"}}' https://10.144.24.54:31943/v3/projects

响应

响应样例

```
{
    "project": {
        "is_domain": false,
        "description": "",
        "links": {
              "self": "10.10.10.10/v3/projects/3de1461665f045ef91ba1efe8121b979"
        },
        "enabled": true,
        "id": "3de1461665f045ef91ba1efe8121b979",
        "parent_id": "d1294857fdf64251994892b344f53e88",
        "domain_id": "d1294857fdf64251994892b344f53e88",
        "name": "region_test1"
    }
}
```

状态码	说明
201	请求成功。
400	参数无效。
401	认证失败。
403	鉴权失败。

状态码	说明
409	项目名重复。

4.4.5 修改项目

功能介绍

该接口提供修改项目信息的功能。

URI

● URI格式

PATCH /v3/projects/{project_id}

● URI参数说明

参数	是否为必 选	类型	说明
id	是	String	项目的id值。

请求

● Request Header参数说明

参数	是否为必选	类型	说明
Content-Type	是	String	文本类型及编码方式。 该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	有效的token(当前租户下具有 secu_admin权限的用户token)。

● Request Body参数说明

属性	是否为必选	类型	说明
name	否	String	项目名,必须以存在的区域ID开 头,长度小于等于64。 示例: {region}_test2
description	否	String	项目描述,长度小于等于255。

● 请求示例

curl -i -k -H "X-Auth-Token:\\$token" -H 'Content-Type:application/json;charset=utf8' -X PATCH -d '{"project":\"name":"region_test2", "description":"test_project_desc"}}' https://
10.144.24.54:31943/v3/projects/23da5961c8214f5caf701c27d9703959

响应消息

响应示例

状态码

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	鉴权失败。
409	项目名称重复。

4.4.6 查询指定项目的信息

功能介绍

该接口提供根据Project ID获取Project的详细信息。

URI

● URI格式

GET /v3/projects/{project_id}

● URI参数说明

参数	是否必选	类型	描述
project_id	是	String	指定的项目ID。

请求

● Request Header参数说明

参数	是否必选	类型	描述
X-Auth-Token	是	String	有效的token(当 前租户的 token)。
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。

● 请求示例

获取project.id=619d3e78f61b4be68bc5aa0b59edcf7b的project信息

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.64.23.150:31943/v3/projects/619d3e78f61b4be68bc5aa0b59edcf7b

响应

响应示例

```
"project": {
    "is_domain": false,
    "description": "",
    "links": {
        "self": "None/v3/projects/2e93d63d8d2249f5a4ac5e2c78586a6e"
    },
    "enabled": true,
    "id": "2e93d63d8d2249f5a4ac5e2c78586a6e",
    "id": "2e93d63d8d2249f5a4ac5e2c78586a6e",
    "parent_id": "44c0781c83484eb9a4a5d4d233522cea",
    "domain_id": "44c0781c83484eb9a4a5d4d233522cea",
    "name": "MOS"
}
```

状态码	说明
200	请求成功
400	请求错误。
401	认证失败。
403	鉴权失败。
404	资源不存在。
500	内部服务错误。

4.4.7 设置指定项目的状态

功能介绍

该接口用于设置指定项目的状态,项目的状态包括正常、冻结。

URI

● URI格式

PUT /v3-ext/projects/{project_id}

● URI参数说明

参数	是否必选	类型	说明
project_id	是	String	项目ID。

● Request Body参数说明

属性	是否为必选	类型	说明
status	是	String	项目的状态信息,范围 ['suspended', 'normal']。

∭说明

- status值为'suspended'时,会将project设置为冻结状态;
- status值为 'normal' 时,会将project设置为正常 (解冻) 状态。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	String	有效的token(当前租户下具有 secu_admin的用户token)。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -X "X-Auth-Token:\$token" -X PUT -d '{"project": {"status":"suspended"}}'https://
10.145.93.56:31943/v3-ext/projects/5c9f5525d9d24c5bbf91e74d86772029

响应

无响应体。

状态码	说明
204	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
500	内部服务错误。
503	服务不可用。

4.4.8 查询指定项目信息及状态

功能介绍

该接口用于查询指定项目的详细信息,包含项目的状态。

URI

- URI格式 GET /v3-ext/projects/{project_id}
- URI参数说明

参数	是否必选	类型	说明
project_id	是	String	项目ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	String	有效的token(当前租户下具有 secu_admin的用户token)。

● 请求样例

 $curl \ -i \ -k \ -H \ 'Accept:application/json' \ -H \ 'Content-Type:application/json; charset=utf8' \ -X \ "X-Auth-Token: \$token" \ -X \ GET \ https://10.145.93.56:31943/v3-ext/projects/5c9f5525d9d24c5bbf91e74d86772029$

响应

● Response Body参数说明

参数	是否必选	类型	说明
project	是	Object	项目信息。

● project格式说明

参数	是否必选	类型	说明
description	是	String	项目描述。
id	是	String	项目ID。
domain_id	是	String	项目所在domain的id。
name	是	String	项目名称。
is_domain	是	Boolean	是否为domain。
enabled	是	Boolean	项目是否可用。
parent_id	是	String	项目的父ID。
status	是	String	项目的状态。
suspended_time	否	String	项目的冻结时间。

● 响应样例

```
{
    "project": {
        "is_domain": false,
        "description": "",
        "enabled": true,
        "id": "ee65ca70d3cf43aaalea6492ce15f289",
        "parent_id": "9041929bcc6e4bfe85add4e7b96ffdd7",
        "domain_id": "398998b5392f4150ad48fe456d6de4f1",
        "name": "cn-north-1_test1",
        "status": "suspended",
        "suspended_time": "2017-08-17T02:50:23.000000"
    }
}
```

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
500	内部服务错误。

状态码	说明
503	服务不可用。

4.5 委托

4.5.1 创建委托

功能介绍

该接口提供创建委托关系的功能。

URI

● URI格式 POST /v3.0/OS-AGENCY/agencies

请求消息

● Request Header参数说明

参数	必选/可选	类型	描述
Content-Type	必选	String	application/ json;charset=utf8
X-Auth-Token	必选	String	已认证的拥有 secu_admin权限的 token。

● Request Body参数说明

参数	是否为 必选	类型	说明
name	是	String	委托的名称,长度不大于64位。
domain_id	是	String	当前租户的ID。
trust_domain_id	至少一个	String	委托租户的ID。当trust_domain_id 和trust_domain_name都存在时,以 trust_domain_name为主。
trust_domain_name		String	委托租户的名称。
description	否	String	委托的描述,长度不大于255位。

□ 说明

请求体中的trust_domain_id和trust_domain_name两者二选一,即必须有一个存在,两者都存在的话,以trust domain name为主。

● 请求示例

```
curl -i -k -H "X-Auth-Token:$token" -H 'Content-Type:application/json;charset=utf8' -X POST -
d'{"agency": {"name": "exampleagency", "domain_id":
"0ae9c6993a2e47bb8c4c7a9bb8278d61", "trust_domain_id":
"35d7706cedbc49a18df0783d00269c20", "trust_domain_name": "exampledomain", "description":
"testsfdas"}}' https://sample.domain.com/v3.0/OS-AGENCY/agencies
```

响应消息

● Response Body参数说明

参数	必选/可选	类型	描述
agency	是	JSONObject	委托对象。

● agency格式说明

参数	必选/可选	类型	描述
id	是	String	委托的ID。
name	是	String	委托的名称。
domain_id	是	String	当前租户的ID。
trust_domain_id	是	String	委托租户的ID。
description	是	String	委托的描述。
duration	是	String	委托的期限,默认是永 久。
expire_time	是	String	委托的到期时间。
create_time	是	String	委托的创建时间。

● 响应示例(请求成功)

```
{
    "agency" : {
        "description" : "testsfdas",
        "trust_domain_id" : "35d7706cedbc49a18df0783d00269c20",
        "id" : "c1a06ec7387f430c8122d6f336c66dcf",
        "duration" : null,
        "create_time" : "2017-01-06T05:56:09.738212",
        "expire_time" : null,
        "domain_id" : "0ae9c6993a2e47bb8c4c7a9bb8278d61",
        "name" : "exampleagency"
    }
}
```

● 响应样例(请求失败)

```
{
    "error": {
        "message": "'name' is a required property",
        "code": 400,
        "title": "Bad Request"
```

状态码	说明
201	请求成功。
400	参数错误。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
409	委托已经存在。
500	内部服务错误。

4.5.2 获取指定委托的详细信息

功能介绍

该接口用来获取指定委托的详细信息。

URI

● URI格式

GET /v3.0/OS-AGENCY/agencies/{agency_id}

● URI参数说明

参数	是否为必选	类型	说明
agency_id	是	String	委托的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	己认证的拥有secu_admin权限的token。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3.0/0S-AGENCY/agencies/2809756f748a46e2b92d58d309f67291

响应消息

● Response Body参数说明

参数	必选/可选	类型	描述
agency	是	JSONObject	委托对象。

● agency格式说明

参数	必选/可选	类型	描述
id	是	String	委托的ID。
name	是	String	委托的名称。
domain_id	是	String	当前租户的ID。
trust_domain_id	是	String	委托租户的ID。
trust_domain_nam e	是	String	委托租户的名称。
description	是	String	委托的描述。
duration	是	String	委托的期限,默认是永 久。
expire_time	是	String	委托的到期时间。
create_time	是	String	委托的创建时间。

● 响应示例(请求成功)

```
{
    "agency" : {
        "description" : " testsfdas ",
        "trust_domain_id" : "3ebe1024db46485cb02ef08d3c348477",
        "trust_domain_name" : "exampledomain",
        "id" : "c1a06ec7387f430c8122d6f336c66dcf",
        "duration" : null,
        "create_time" : "2017-01-06T05:56:09.738212",
        "expire_time" : null,
        "domain_id" : "0ae9c6993a2e47bb8c4c7a9bb8278d61",
        "name" : "exampleagency"
    }
}
```

● 响应样例 (请求失败)

```
{
    "error": {
        "message": "Could not find agency: 2809756f748a46e2b92d58d309f67291",
        "code": 404,
        "title": "Not Found"
    }
}
```

状态码	说明
200	请求成功。
401	认证失败。
403	鉴权失败。
404	不存在对应的委托。
500	内部服务错误。

4.5.3 查询指定条件下的委托列表信息

功能介绍

该接口用来获取指定条件下的委托列表信息。

URI

● URI格式

GET /v3.0/OS-AGENCY/agencies {?domain_id,name}

● 查询参数说明

参数	是否为必选	类型	说明
domain_id	是	String	当前租户的ID。
name	否	String	委托的名称。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3.0/OS-AGENCY/agencies?domain_id=0ae9c6993a2e47bb8c4c7a9bb8278d61

响应消息

● Response Body参数说明

参数	必选/可选	类型	描述
agencies	是	JSONArray	委托列表。

● agency格式说明

参数	必选/可选	类型	描述
id	是	String	委托的ID。
name	是	String	委托的名称。
domain_id	是	String	当前租户的ID。
trust_domain_id	是	String	委托租户的ID。
trust_domain_nam e	是	String	委托租户的名称。
description	是	String	委托的描述。
duration	是	String	委托的期限,默认是永 久。
expire_time	是	String	委托的到期时间。
create_time	是	String	委托的创建时间。

● 响应示例(请求成功)

● 响应样例(请求失败)

```
{
  "error": {
    "message": "You are not authorized to perform the requested action: identity:list_agencies",
    "code": 403,
    "title": "Forbidden"
}
```

状态码	说明
200	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.4 修改委托

功能介绍

该接口提供修改委托的信息,包括租户的trust_domain_id、description、trust_domain_name字段。

URI

● URI格式

PUT /v3.0/OS-AGENCY/agencies/{agency_id}

● URI参数说明

参数	是否为必选	类型	说明
agency_id	是	String	委托的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	已认证的拥有secu_admin权限的 token。

● Request Body参数说明

属性	是否为必选	类型	说明
trust_domain_id	否	String	委托租户的ID,且委托租户 必须已存在。

属性	是否为必选	类型	说明
trust_domain_nam e	否	String	委托租户的名称,且委托租 户必须已存在。
description	否	String	委托的描述。

∭说明

请求体中的trust_domain_id和trust_domain_name两者二选一,即有一个存在即可,两者都存在的话,以trust_domain_name为主。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X PUT -d '{"agency": {"trust_domain_id": "35d7706cedbc49a18df0783d00269c20", "trust_domain_name": "exampledomain", "description": "111111"}}' https://sample.domain.com/v3.0/0S-AGENCY/agencies/2809756f748a46e2b92d58d309f67291

响应消息

● Response Body参数说明

参数	必选/可选	类型	描述
agency	是	JSONObject	委托对象。

● agency格式说明

参数	必选/可选	类型	描述
id	是	String	委托的ID。
name	是	String	委托的名称。
domain_id	是	String	当前租户的ID。
trust_domain_id	是	String	委托租户的ID。
description	是	String	委托的描述。
duration	是	String	委托的期限,默认是永 久。
expire_time	是	String	委托的到期时间。
create_time	是	String	委托的创建时间。

● 响应示例(请求成功)

```
{
    "agency" : {
        "description" : " testsfdas ",
        "trust_domain_id" : "3ebe1024db46485cb02ef08d3c348477",
        "id" : "c1a06ec7387f430c8122d6f336c66dcf",
        "duration" : null,
        "create_time" : "2017-01-06T05:56:09.738212",
        "expire_time" : null,
        "domain_id" : "0ae9c6993a2e47bb8c4c7a9bb8278d61",
```

```
"name" : "exampleagency"
}
}
```

● 响应样例 (请求失败)

```
{
  "error": {
    "message": "TrustDomainNotFound",
    "code": 404,
    "title": "Not Found"
}
```

状态码

状态码	说明
200	请求成功。
400	参数错误。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.5 删除委托

功能介绍

该接口用来删除委托。

URI

● URI格式

DELETE /v3.0/OS-AGENCY/agencies/{agency_id}

● URI参数说明

参数	是否为必选	类型	说明
agency_id	是	String	委托的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X DELETE https://sample.domain.com/v3.0/OS-AGENCY/agencies/2809756f748a46e2b92d58d309f67291

响应消息

● 响应样例(请求失败)

```
{
    "error": {
        "message": "Could not find agency: 2809756f748a46e2b92d58d309f67291",
        "code": 404,
        "title": "Not Found"
    }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.6 通过项目给委托赋予权限

功能介绍

该接口用来通过项目给委托赋予权限。

URI

● URI格式

PUT /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}

● URI参数说明

参数	是否为必选	类型	说明
project_id	是	String	当前租户下的项目ID。
agency_id	是	String	委托的ID。
role_id	是	String	角色对应的ID。

∭说明

请求体中角色id对应的角色名称有黑名单控制,不能是secu_admin、te_agency和以op_开头的名称。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X PUT https://sample.domain.com/v3.0/0S-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff

响应消息

● 响应样例(请求失败)

```
{
    "error": {
        "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
        "code": 404,
        "title": "Not Found"
    }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.7 检查委托在项目上是否有指定权限

功能介绍

该接口用来检查委托在项目上是否有指定权限。

URI

● URI格式

HEAD /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}

● URI参数说明

参数	是否为必选	类型	说明
project_id	是	String	当前租户下的项目ID。
agency_id	是	String	委托的ID。
role_id	是	String	角色对应的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X HEAD https://sample.domain.com/v3.0/0S-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff

响应消息

● 响应样例(请求失败)

```
{
    "error": {
        "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
        "code": 404,
        "title": "Not Found"
    }
}
```

状态码	说明
204	请求成功,有指定权限。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.8 查询委托在项目上具有的权限列表

功能介绍

该接口用来查询委托在项目上具有的权限列表。

URI

● URI格式

GET /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles

● URI参数说明

参数	是否为必选	类型	说明
project_id	是	String	当前租户下的项目ID。
agency_id	是	String	委托的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3.0/0S-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles

响应消息

● Response Body参数说明

参数	是否为必 选	类型	说明
roles	是	Array	role的列表。

● role格式说明

参数	是否为必 选	类型	说明
id	是	String	role的ID。
name	是	String	role的名称。
domain_id	是	String	role所属的租户的ID。
type	是	String	role的显示模式,其中: AX表示在domain层显示; XA表示在 project层显示; AA表示在domain和 project层均显示; XX表示在domain和 project层均不显示。
display_name	否	String	role所展示名称。
catalog	否	String	role所在的目录。
policy	否	Dict	role的具体策略。
description	否	String	role的描述。

● 响应样例

● 响应样例(请求失败)

```
{
  "error": {
    "message": "You are not authorized to perform the requested action:
identity:list_domain_grants",
    "code": 403,
    "title": "Forbidden"
  }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.9 通过项目删除委托的权限

功能介绍

该接口用来通过项目信息删除委托的权限。

URI

● URI格式

DELETE /v3.0/OS-AGENCY/projects/{project_id}/agencies/{agency_id}/roles/{role_id}

● URI参数说明

参数	是否为必选	类型	说明
project_id	是	String	当前租户下的项目ID。
agency_id	是	String	委托的ID。
role_id	是	String	角色对应的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type: application/json; charset=utf8' -X DELETE https://sample.domain.com/v3.0/OS-AGENCY/projects/0945241c5ebc4660bac540d48f2a2c14/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff

响应消息

● 响应样例 (请求失败)

```
{
    "error": {
        "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
        "code": 404,
        "title": "Not Found"
    }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.10 通过租户给委托赋予权限

功能介绍

该接口用来通过租户信息给委托赋予权限。

URI

● URI格式

PUT /v3.0/OS-AGENCY/domains/{domain id}/agencies/{agency id}/roles/{role id}

● URI参数说明

参数	是否为必选	类型	说明
domain_id	是	String	当前租户的ID。
agency_id	是	String	委托的ID。
role_id	是	String	角色对应的ID。

∭说明

请求体中角色id对应的角色名称有黑名单控制,不能是secu_admin、te_agency和以op_开头的名称。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X PUT https://sample.domain.com/v3.0/0S-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff

响应消息

● 响应样例(请求失败)

```
{
    "error" : {
        "message" : "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
        "code" : 404,
        "title" : "Not Found"
     }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.11 检查委托在租户上是否有指定权限

功能介绍

该接口用来检查委托在租户上是否具有指定权限。

URI

● URI格式

HEAD /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}

● URI参数说明

参数	是否为必选	类型	说明
domain_id	是	String	当前租户的ID。
agency_id	是	String	委托的ID。
role_id	是	String	角色对应的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type: application/json; charset=utf8' -X HEAD https://sample.domain.com/v3.0/0S-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff

响应消息

● 响应样例(请求失败)

```
{
    "error": {
        "message": "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
        "code": 404,
        "title": "Not Found"
    }
}
```

状态码	说明
204	请求成功,具有指定权限。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.12 查询委托在租户上具有的权限列表

功能介绍

该接口用来查询委托在租户上具有的权限列表。

URI

● URI格式

GET /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles

● URI参数说明

参数	是否为必选	类型	说明
domain_id	是	String	当前租户的ID。
agency_id	是	String	委托的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://sample.domain.com/v3.0/0S-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles

响应消息

● Response Body参数说明

参数	是否为必 选	类型	说明
roles	是	Array	role的列表。

● role格式说明

参数	是否为必 选	类型	说明
id	是	String	role的ID。
name	是	String	role的名称。
domain_id	是	String	role所属的租户的ID。
type	是	String	role的显示模式,其中: AX表示在domain层显示; XA表示在 project层显示; AA表示在domain和 project层均显示; XX表示在domain和 project层均不显示。
display_name	否	String	role所展示名称。
catalog	否	String	role所在的目录。
policy	否	Dict	role的具体策略。
description	否	String	role的描述。

● 响应样例

● 响应样例(请求失败)

```
{
  "error": {
    "message": "You are not authorized to perform the requested action:
identity:list_domain_grants",
    "code": 403,
    "title": "Forbidden"
  }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.5.13 通过租户删除委托的权限

功能介绍

该接口用来通过租户信息删除委托的权限。

URI

● URI格式

DELETE /v3.0/OS-AGENCY/domains/{domain_id}/agencies/{agency_id}/roles/{role_id}

● URI参数说明

参数	是否为必选	类型	说明
domain_id	是	String	当前租户的ID。
agency_id	是	String	委托的ID。
role_id	是	String	角色对应的ID。

请求消息

● Request Header参数说明

参数	是否为必 选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	当前租户下具有secu_admin权限的 token。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X DELETE https://sample.domain.com/v3.0/0S-AGENCY/domains/b32d99a7778d4fd9aa5bc616c3dc4e5f/agencies/37f90258b820472bbc8a0f4f0bfd720d/roles/0f3a2d418ed747fa8be46e92757be9ff

响应消息

● 响应样例(请求失败)

```
{
    "error" : {
        "message" : "Could not find role: 0f3a2d418ed747fa8be46e92757be9ddff",
        "code" : 404,
        "title" : "Not Found"
     }
}
```

状态码

状态码	说明
204	请求成功。
401	认证失败。
403	鉴权失败。
404	未找到相应的资源。
500	内部服务错误。

4.6 区域

4.6.1 查询区域列表

功能介绍

该接口用于查询区域列表。

URI

URI格式

GET /v3/regions

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的token。 如果token中没有 私有region标签, 则查询结果不返 回私有region。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/regions

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
links	是	Dict	区域的链接,包含 next,previous,self。
regions	是	List	区域列表。

● services格式说明

参数	是否必选	类型	说明
description	是	String	区域描述。
parent_region_id	是	String	区域的父区域 ID。
id	是	String	区域id。
locales	是	Dict	区域名。
type	是	String	区域类型。
links	是	Dict	区域的链接。

● 响应样例(响应成功)

```
"links": {
                 "self": "None/v3/regions/1500365963661574434"
           "type": "private",
"id": "1500365963661574434",
           "locales": {
                "pt-br": "POTRGA",
"en-us": "ENGLISH",
"es-es": "ESPANIAR",
"zh-cn": "region_name"
           "parent_region_id": null,
            "description": "",
           "links": {
                  "self": "None/v3/regions/500017826026667755"
           },
"type": "private",
"id": "500017826026667755",
           "locales": {
    "pt-br": "POTRGA",
    "en-us": "ENGLISH",
    "es-es": "ESPANIAR",
    """ "stain nam
                  "zh-cn": "region_name1"
      },
           "parent_region_id": null,
"description": "",
            "links": {
                  "self": "None/v3/regions/region_name"
           },
"type": "public",
"id": "Dec-test2",
           "locales": {
    "zh-cn": "region_name2",
    "en-us": "region_name2"
     },
            "parent_region_id": null,
            "links": {
                 "self": "None/v3/regions/Dec-test1112244"
           },
"id": "Dec-test1112244",
           "locales": {
                 "zh-cn": "测试区1_test01",
                 "en-us": "testregionl",
"es-es": "testregionl",
"pt-br": "testregionl"
           },
"description": ""
],
"links": {
      "self": "None/v3/regions",
      "previous": null,
      "next": null
```

状态码	说明
200	请求成功。

状态码	说明
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.6.2 查询区域详情

功能介绍

该接口用于查询区域详情。

URI

- URI格式 GET /v3/regions/{region_id}
- 参数说明

参数	是否必选	类型	说明
region_id	是	String	指定区域ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/regions/test-pusb99999991

响应

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.7 权限

4.7.1 查询权限列表

功能介绍

该接口提供查询权限列表。

URI

● URI格式 GET /v3/roles

● URL参数说明

属性	是否必选	类型	说明
name	否	String	role的名称。
domain_id	否	String	Domain的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth或 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://10.22.44.158:31943/v3/roles?name=readonly

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
links	是	Dict	role的链接,包含 next,previous,self。
roles	是	Array	role的列表。

● role格式说明

参数	是否必选	类型	说明
id	是	String	role的ID。
links	是	Dict	role的链接。
name	是	String	role的名称。
domain_id	是	String	role所属Domain的ID。

参数	是否必选	类型	说明
type	是	String	role的显示模式,其中:
			AX表示在domain层显示;
			XA表示在project层显示;
			AA表示在domain和project层均显示;
			XX表示在domain和project层均不显示。
display_name	否	String	role所展示的名称。
catalog	否	String	role所在的目录。
policy	否	Dict	role的具体策略。
description	否	String	role的描述。

● 响应样例

```
"previous": null,
"next": null
},
"roles": [
   {
     "display_name": "Tanent Guest",
"description": "Tanent Guest",
     "links": {
    "self": "www.example.com/v3/roles/19bb93eec4ca4f08aefdc02da76d8f3c"
     },
  "domain_id": null,
  "catalog": "BASE",
  "policy": {
        "Version": "1.0",
        "Statement": [
             "Action": [
"::Get",
"::List"
             ],
"Effect": "Allow"
             "Action": [
                "identity:*"
             "Effect": "Deny"
        ]
     },
"id": "19bb93eec4ca4f08aefdc02da76d8f3c",
"type": "AA",
"name": "readonly"
]
```

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。

4.7.2 查询权限的详细信息

功能介绍

该接口提供查询权限详情。

URI

- URI格式 GET /v3/roles/{role_id}
- URL参数说明

属性	是否必选	类型	说明
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth或 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H "X-Auth-Token:\$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://10.22.44.158:31943/v3/roles/19bb93eec4ca4f08aefdc02da76d8f3c

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
role	是	Dict	role的详细信息。

● role格式说明

参数	是否必选	类型	说明
domain_id	是	String	role所属Domain的ID。
id	是	String	role的ID。
links	是	Dict	role的链接。
name	是	String	role的名称。
type	是	String	role的显示模式,其中: AX表示在domain层显示; XA表示在project层显示; AA表示在domain和project层均显示; XX表示在domain和project层均不显示。
display_name	否	String	role所展示的名称。
catalog	否	String	role所在的目录。
policy	否	Dict	role的具体策略。
description	否	String	role的描述。

● 响应样例

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.7.3 查询租户中用户组的权限

功能介绍

该接口提供查询租户下用户组的权限的功能。

URI

- URI格式
 GET /v3/domains/{domain_id}/groups/{group_id}/roles
- URL参数说明

属性	是否必选	类型	说明
domain_id	是	String	用户组所属Domain的ID。
group_id	是	String	用户组的ID。

请求

● Request Header参数说明

参数	是否 必选	类型	说明
Content- Type	是	Stri ng	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	Stri ng	有效的token(具有op_auth权限或当前租户下具有 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type: application/json; charset=utf8' -X GET https://10.22.44.158: 31943/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
links	是	Dict	Domain中指定用户组的role的链接, 包含next,previous,self。
roles	是	Array	Domain中指定用户组的role。

● role参数说明

参数	是否必选	类型	说明
id	是	String	Domain中指定用户组的role的ID。
links	是	Dict	role的链接。
name	是	String	role的名称。
domain_id	是	String	role所属Domain的ID。
type	是	String	role的显示模式,其中: AX表示在domain层显示; XA表示在project层显示; AA表示在domain和project层均显示; XX表示在domain和project层均不显示。
display_name	否	String	role所展示的名称。
catalog	否	String	role所在的目录。
policy	否	Dict	role的具体策略。
description	否	String	role的描述。

● 响应样例

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	鉴权失败。

4.7.4 查询项目对应的用户组的权限

功能介绍

该接口提供查询某个项目对应的指定用户组的权限。

URI

URI格式
 GET /v3/projects/{project id}/groups/{group id}/roles

● URL参数说明

属性	是否必选	类型	说明
project_id	是	String	Project的ID。
group_id	是	String	用户组的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限或当前 租户下具有secu_admin权限的用 户token)。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X GET https://10.22.44.158:31943/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
links	是	Dict	role的链接,包含 next,previous,self。
roles	是	Array	role列表。

● role参数说明

参数	是否必选	类型	说明
id	是	String	role的ID。
links	是	Dict	role的链接。
name	是	String	role的名称。
domain_id	是	String	role所属Domain的ID。

参数	是否必选	类型	说明
type	是	String	role的显示模式,其中:
			AX表示在domain层显示;
			XA表示在project层显示;
			AA表示在domain和project层均显
			示;
			XX表示在domain和project层均不显示。
display_name	否	String	role所展示的名称。
catalog	否	String	role所在的目录。
policy	否	Dict	role的具体策略。
description	否	String	role的描述。

● 响应样例

```
"links": {
          "self": " www.example.com/v3/projects/3a4cd4d559d8492bbe7bd355643f9763/groups/
728da352c017480f80b5a96beb15f0e6/roles",
          "previous": null,
"next": null
     },
"roles": [
               "catalog": "BASE",
"display_name": "Guest",
"name": "readonly",
"links": {
                     "self": " www.example.com/v3/roles/13d132b7856945788f6df7eb3ed5c35e"
               },
"policy": {
                     "Version": "1.0",
                    "Statement": [
                               "Action": [
                                    "*:*:Get*",
"*:*:List*"
                               "Effect": "Allow"
                         },
{
                               "Action": [
                                    "identity:*"
                               "Effect": "Deny"
                    ]
               },
"domain_id": null,
               "type": "AA",
"id": "13d132b7856945788f6df7eb3ed5c35e",
               "description": "Guest"
          },
               "catalog": "BASE",
"display_name": "Tenant Administrator",
"name": "te_admin",
"links": {
```

状态码	说明
200	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。

4.7.5 租户的用户组授权

功能介绍

该接口提供租户中用户组增加权限的功能。

URI

● URI格式

PUT /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}

● URL参数说明

属性	是否必选	类型	说明
domain_id	是	String	用户组所属Domain的ID。
group_id	是	String	用户组的ID。
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth、 op_service权限或当前租户下具有 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type: application/json; charset=utf8' -X PUT https://10.22.44.158: 31943/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	鉴权失败。
404	找不到资源。
409	资源冲突。

4.7.6 为项目对应的用户组授权

功能介绍

该接口提供项目对应用户组增加权限的功能。

URI

- URI格式PUT /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}
- URL参数说明

属性	是否必选	类型	说明
project_id	是	String	Project的ID。
group_id	是	String	用户组的ID。
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	有效的token(op_auth、op_service权限或当前租户下具有secu_admin权限的用户token)。

● 请求样例

curl - i - k -H "X-Auth-Token: \$token" -H 'Content-Type: application/json; charset=utf8' -X PUT https://10.22.44.158:31943/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。
409	资源冲突。

4.7.7 删除项目对应的用户组的权限

功能介绍

该接口提供删除项目对应用户组权限的功能。

URI

● URI格式

DELETE /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}

● URL参数说明

属性	是否必选	类型	说明
project_id	是	String	Project的ID。
group_id	是	String	用户组的ID。
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth-Token	是	String	有效的token(op_auth权限或当前 租户下具有secu_admin权限的用 户token)。

● 请求样例

 $curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type: application/json; charset=utf8' -X DELETE \ https://10.22.44.158: 31943/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2$

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	鉴权失败。
404	找不到资源。

4.7.8 删除租户中用户组的权限

功能介绍

该接口提供删除某租户中指定用户组权限的功能。

URI

● URI格式

DELETE /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}

● URL参数说明

属性	是否必选	类型	说明
domain_id	是	String	用户组所属Domain的ID。
group_id	是	String	用户组的ID。
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否 必选	类型	说明
Content- Type	是	String	该字段内容填为"application/json;charset=utf8"。
X-Auth- Token	是	String	有效的token(op_auth权限或当前租户下具有 secu_admin权限的用户token)。

● 请求样例

 $curl \ -i \ -k \ -H \ "X-Auth-Token: \$token" \ -H \ 'Content-Type: application/json; charset=utf8' \ -X \ DELETE \ https://10.22.44.158: 31943/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2$

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。

状系	态码	说明
404	4	找不到资源。

4.7.9 查询租户中用户组是否包含权限

功能介绍

该接口提供检查某租户中指定用户组是否包含特定权限的功能。

URI

● URI格式

HEAD /v3/domains/{domain_id}/groups/{group_id}/roles/{role_id}

● URL参数说明

属性	是否必选	类型	说明
domain_id	是	String	用户组所属Domain的ID。
group_id	是	String	用户组的ID。
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否 必选	类型	说明
Content- Type	是	Stri ng	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	Stri ng	有效的token(op_auth权限或当前租户下具有 secu_admin权限的用户token)。

● 请求样例

curl - i - k - H "X-Auth-Token: \$token" - H 'Content-Type: application/json; charset=utf8' - X HEAD https://10.22.44.158:31943/v3/domains/d54061ebcb5145dd814f8eb3fe9b7ac0/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2

响应

无响应体。

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	鉴权失败。
404	找不到资源。

4.7.10 查询项目对应的用户组是否包含权限

功能介绍

该接口提供查询项目对应用户组是否包含权限的功能。

URI

● URI格式

HEAD /v3/projects/{project_id}/groups/{group_id}/roles/{role_id}

● URL参数说明

属性	是否必选	类型	说明
project_id	是	String	Project的ID。
group_id	是	String	用户组的ID。
role_id	是	String	role的ID。

请求

● Request Header参数说明

参数	是否 必选	类型	说明
Content- Type	是	Stri ng	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	Stri ng	有效的token(op_auth权限或当前租户下具有 secu_admin权限的用户token)。

● 请求样例

curl -i -k -H "X-Auth-Token: \$token" -H 'Content-Type:application/json;charset=utf8' -X HEAD https://10.22.44.158:31943/v3/projects/073bbf60da374853841cf6624c94de4b/groups/47d79cabc2cf4c35b13493d919a5bb3d/roles/e62d9ba0d6a544cd878d9e8a4663f6e2

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	参数无效。
401	认证失败。
403	没有操作权限。
404	找不到资源。

4.8 Token

4.8.1 获取用户 Token

功能介绍

该接口用来获取用户Token,可以通过用户名/密码的方式进行认证。

□ 说明

- Token的有效期为24小时,需要同一个Token鉴权时,可以先缓存起来,避免频繁调用。
- 该接口属于服务间接口,提供了锁定机制用于防止暴力破解,服务调用时,请做好用户名密码的充分校验。

URI

URI格式

POST /v3/auth/tokens

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content- Type	是	String	该字段内容填为 "application/json;charset=utf8"。

● Request Body参数说明

参数	是否必选	类型	说明
methods	是	String Array	该字段内容为"password"。
user	是	Json Object	示例: "user": { "name": "username", "password": "password", "domain": { "name": "domainname" } } domainname: 用户所属的企业账户名称。 username: 用户名称。 password: 用户登录时的密码。
scope	否	Json Object	token的作用范围。支持domain、project,但两者不能设置在同一层级上。 示例1("name"为企业账户名称,此处以"examplename"为例): "scope":{ "domain":{ "name":"examplename" } i 该示例表示本token仅能访问examplename 企业账户下的资源。 示例2: "scope":{

● 请求样例

获取用户名为exampleuser,登录密码为Examplepassword123,所在domain名为exampledomain的token。

响应

● Response Header参数说明

参数	是否必选	类型	描述
X-Subject- Token	是	String	签名后的token。

● Token格式说明

参数	是否必选	类型	描述
methods	是	Json Array	获取token的方式。
expires_at	是	String	token超时时间。
issued_at	是	String	token产生时间。
user	是	Json Object	示例: "user": { "name": "username", "id": "userid", "domain": { "name": "domainname", "id": "domainid" } } username: 用户名称。 userid: 用户id。 domainname: 用户所属的企业账户名称。 domainid: 用户所属的企业账户的域id。

参数	是否必选	类型	描述
domain	否	Json Object	根据请求中的scope,判断是否返回该字段。 示例: "domain": {
project	否	Json Object	根据请求中的scope,判断是否返回该字段。 示例: "project": { "name": "projectname", "id": "projectid", "domain": { "name": "domainname", "id": "domainid" } } projectname: project名称。 projectid: project的id。 domainname: poject所属的企业账户名称。 domainid: project所属的企业账户的域id。
catalog	是	Json Array	endpoints相关信息。 示例: "catalog": [{ "type": "identity", "id": "1331e5cff2a74d76b03da1225910e31d", "name": "iam", "endpoints": [{ "url": "www.example.com/v3", "region": "*", "interface": "public", "id": "089d4a381d574308a703122d3ae738e9" }] }]
roles	是	Json Object	角色数组。 示例: "roles":[{

● 响应样例

```
Response Header中存储信息为:
X-Subject-Token: MIIDkgYJKoZIhvcNAQcCoIIDgzCCA38CAQExDTALBg1ghkgBZQMEAgEwgXXXXX...
Response Body中存储信息为:
   "token" : {
      "methods" : ["password"],
"expires_at" : "2015-11-09T01:42:57.527363Z",
"issued_at" : "2015-11-09T00:42:57.527404Z",
       "user" : {
           "domain" : {
          "id": "default",
"name": "exampledomain"
          },
"id" : "ee4dfb6e5540447cb3741905149d9b6e",
"name" : "exampleuser"
     "domain" : {
    "name" : "exampledomain",
    "id" : "default"
     },
    "catalog": [{
        "type": "identity",
        "id": "1331e5cff2a74d76b03da1225910e31d",
        "io": "iam",
        ""
             "id": "1331e5cff2a74d76b03da12259
"name": "iam",
"endpoints": [{
    "url": "www.example.com/v3",
    "region": "*",
    "region_id": "*",
    "interface": "public",
    "id": "080d4c22ld57/208c703
                       "id": "089d4a381d574308a703122d3ae738e9"
           }]
     }, {
"name" : "role2",
"id" : "roleid2"
```

状态码

状态码	说明	
201	请求成功。	
400	请求错误。	
401	认证失败。	
403	鉴权失败。	
404	找不到资源。	
500	内部服务错误。	
503	服务不可用。	

4.8.2 通过 assume_role 方式获取用户 token

功能介绍

该接口用来获取用户token,可以通过assume role的方式进行认证。

□□说明

token的有效期为24小时,需要同一个token鉴权时,可以先缓存起来,避免频繁调用。

URI

URI格式

POST /v3/auth/tokens

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	已认证的拥有te_agency权限的token。

● Request Body参数说明

参数	是否为必选	类型	说明
methods	是	String Array	该字段内容填为"assume_role"。
domain_nam	是	String	指定租户名。
xrole_name	是	String	该字段内容填为委托名称。
restrict	否	Json Object	限定的权限集,填写roles的信息。
roles	否	List Object	权限信息。
scope	否	Json Object	token所属范围信息。

□说明

上述参数信息的获取方式请参考"相关参数信息获取"。

● 请求样例:

curl -i -k -H 'X-Auth-Token: \$token' -H 'Content-Type:application/json; charset=utf8' -X POST -d '{"auth": {"identity": {"methods": ["assume_role"], "assume_role": {"domain_name": "exampledomain", "xrole_name": "op_service", "restrict": {"roles": ["secu_admin"]}}}, "scope": {"domain": {"name": "exampledomain"}}}' https://sample.domain.com/v3/auth/tokens

响应

● Response Header参数说明

参数	是否必选	类型	描述
X-Subject- Token	是	String	签名后的token。

● Token格式说明

参数	是否必选	类型	描述
methods	是	Json Array	获取token的方式。
expires_at	是	String	token超时时间。
issued_at	是	String	token产生时间。
user	是	Json Object	示例: "user": { "name": "username", "id": "userid", "domain": { "name": "domainname", "id": "domainid" } } username: 用户名称。 userid: 用户id。 domainname: 用户所属的企业账户名称。 domainid: 用户所属的企业账户的域id。
domain	否	Json Object	根据请求中的scope,判断是否返回该字段。 示例: "domain": {

参数	是否必选	类型	描述
project	否	Json Object	根据请求中的scope,判断是否返回该字段。 示例: "project": { "name": "projectname", "id": "projectid", "domain": { "name": "domainname", "id": "domainid" } } projectname: project名称。 projectid: project的id。 domainname: poject所属的企业账户名称。 domainid: project所属的企业账户的域id。
catalog	是	Json Array	endpoints相关信息。 示例: "catalog": [{ "type": "identity", "id": "1331e5cff2a74d76b03da1225910e31d", "name": "iam", "endpoints": [{ "url": "www. example. com/v3", "region": "*", "region_id": "*", "interface": "public", "id": "089d4a381d574308a703122d3ae738e9" }] }]
roles	是	Json Object	角色数组。 示例: "roles": [{

● 响应样例

Response Header中存储信息为:

X-Subject-Token:

 $\label{thm:milegy_JkozInvcNaQcCoIIE5zCCBOMCaQexDTALBglghkgBZQMEAgEwggNEBgkqhkiG9w0BBwGgggM1BIIDMXsidG9rZW4iOnsibWV0aG9kcyI6WyJod19hc3N1bWVfcm9sZSJdLCJpc3N1ZWRfYXQi0iIyMDE3LTA1LTE4VDEx0jQ00jA1Lj1zMjAwMFoiLCJ1eHBpcmVzX2F0IjoiMjAxNy0wNS0x0VQxMTo0NDowNS4yMzIwMDBaIiwidXNlciI6eyJpZCI6IjkzZTEyZWNkYWQ2ZjRhYmQ4NDk20Dc0MWRhZjVjNmEzIiwibmFtZSI6ImppeGlhbmcxL29wX3NlcnZpY2UiLCJkb21haW4iOnsiaWQi0iJjZTkyNWM0MmMyNTk0M2J1YmJhMTB1YTY0YWY5MzEwMiIsIm5hbWUi0iJqaXhpYW5nMSJ9fSwiZG9tYWluIjp7ImlkIjoiY2U5MjVjNDJjMjU5NDNiZWJiYTEwZWE2NGFmOTMxMDIiLCJuYW11Ijoiam14aWFuZzEifSwicm9sZXMi01t7ImlkIjoiYzExYzYxMzE5ZjA4NDA0ZWFmOTRm0DAzMGI5ZDM3YmIiLCJuYW11Ijoic2VjdV9hZG1pbiJ9LHsiaWQi0iIwIiwibmFtZSI6Im9wX2x1Z2FjeSJ9LHsiaWQi0iIwIiwibmFtZSI6Im9wX3Jlc3RyaWN0ZWQifSx7ImlkIjoiMCIsIm5hbWUi0iJvcF9nYXR1ZF90YXNzc2cxIn0seyJpZCI6IjAiLCJuYW11Ijoib3BfZ2F0ZWRfdGFzc3NnMiJ9LHsiaWQi0iIwIiwibmFtZ$

 $\label{lem:condition} JuyW111Ijoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11Ijoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11Ijoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11Ijoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11Ijoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11Ijoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11IIjoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11IJoib3Bf22F0ZWRfdGFzc3NnNiJ9XSwiYXNzdW11ZF9ieS16eyJ1c2VyIjp7ImRvbWFpbiI6eyJuYW11IJoib3Bf22F0ZWRfdGFzc3NnNiJ9XWRfdGfzc3NnNiJ9XWRfdGfzc3NnNiJ9XWRfdGfyc3NnNiJ9XWRfdGfyc3NnNiJ9XWRfdGfyc3NnNiJ9XWRfdGfyc3NnNiJ9XWRfdGfyc3NnNiJ9XWRfdffyc3NnJfyc3$ fc2VydmljZSIsImlkIjoiYzFhNzhhODJkODFjNGExOWIwM2JmZTgyZDNhZGQ1ZTUifSwiaWQi0iJjZGViMTU4ZGRhODU0 Y2MzYmFiNzdkODkyNmZmZWNmMyIsIm5hbWUi0iJyb2x1X191c2VyIn19fX0xggGFMIIBgQIBATBcMFcxCzAJBgNVBAYTA 1VTMQ4wDAYDVQQ1DAVVbnN1dDE0MAwGA1UEBwwFVW5zZXQxD.jAMBgNVBAoMBVVuc2V0MRgwFgYDVQQDDA93d3cuZXhhbX BsZS5jb20CAQEwCwYJYIZIAWUDBAIBMAOGCSqGSIb3DQEBAQUABIIBAB4IaWWzCwmqxH +ht4ZNWdvhwZ5NNAH11cCeCjtEvKwu9jOAieeirCx6JjWcbN-uCOkRDHrAny3kj02yAuGW0ndrzXDHQUC8PHzCdk3UPvJnwJdHp7drWIszPfDEM3ThwZTnq27gQY5V0MNFaoJKITtZhK2KcD7zD5GDFMRKF 3pM5vy3naAd4dXCHCfoDsDxHt8XgrYjHEq1KmI-L2ScGBCuZAqQ0L4FwQYQ7W7QDW9ra1m91o30KVdtWI0wb0vaUVx50Wc0PZVoQbJLyuZiX7bu0c71sQ34KegTHwgmWyUNJVfqxQPt8FFhoiXPSiFHVkhz++ioAd8ec-FeABbe8= X-Frame-Options: SAMEORIGIN Response Body中存储信息为: "token": { "methods":["assume_role" "issued_at": "2017-05-18T11:44:05.232000Z", "expires_at": "2017-05-19T11:44:05.232000Z", user": { "id": "93e12ecdad6f4abd84968741daf5c6a3", "name": "exampledomain/op_service", "domain": {
 "id": "ce925c42c25943bebba10ea64af93102", "name": "exampledomain" "domain": { "id": "ce925c42c25943bebba10ea64af93102", "name": "exampledomain" },
"roles": [${\it "id":~"c11c61319f08404eaf94f8030b9d37bb"},$ "name": "secu_admin" "id": "0", "name": "op_legacy" "id": "0",
"name": "op_restricted" "id": "0", "name": "op_gated_tasssg1" "id": "0",
"name": "op_gated_tasssg2" "id": "0", "name": "op_gated_tasssg4" "id": "0", "name": "op_gated_tasssg5" "id": "0",
"name": "op_gated_tasssg6" 'assumed_by": { "user": {

S16Im9wX2dhdGVkX3Rhc3NzZzQifSx7Im1kIjoiMCIsIm5hbWUi0iJvcF9nYXR1ZF90YXNzc2c1In0seyJpZC16IJAiLC

状态码	说明
201	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
500	内部服务错误。
503	服务不可用。

4.9 联邦认证

4.9.1 身份提供商

4.9.1.1 查询身份提供商列表

功能介绍

该接口用于查询身份提供商列表信息。

URI

URI格式

GET /v3/OS-FEDERATION/identity_providers

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	所属租户的 token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.185.190.118:31943/v3/OS-FEDERATION/identity_providers

响应

● Response Body参数说明

参数	是否必选	类型	说明
identity_providers	是	Array	身份提供商列 表。
links	是	Object	身份提供商的资源链接,包含 next, previous, self。

● 响应样例

```
"identity_providers": [
           "description": "Stores ACME identities",
           "enabled": true,
           "id": "ACME",
           "remote_ids": [],
           "links": {
               "protocols": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/
protocols",
               "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME"
       },
           "description": "Stores contractor identities",
           "enabled": false,
           "remote_ids": [],
           "id": "ACME-contractors",
"links": {
               "protocols": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME-
contractors"
   ],
    "links": {
       "next": null,
       "previous": null,
"self": "https://example.com/v3/0S-FEDERATION/identity_providers"
   }
```

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.1.2 查询身份提供商

功能介绍

该接口用于查询身份提供商信息。

URI

● URI格式 GET /v3/OS-FEDERATION/identity_providers/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	身份提供商的 ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	所属租户的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.185.190.118:31943/v3/OS-FEDERATION/identity_providers/ACME

响应

● Response Body参数说明

参数	是否必 选	类型	说明
id	是	String	身份提供商的ID。
descriptio n	是	String	身份提供商的描述信息。
enabled	是	Boolean	身份提供商是否启用,true为启用,false为停用,默认为false。
remote_id	是	Array	身份提供商的远程ID列表。
links	是	Object	身份提供商的资源链接,包含protocols, self。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。

状态码	说明
503	服务不可用。

4.9.1.3 注册身份提供商

功能介绍

该接口用于注册一个身份提供商。

URI

● URI格式

PUT /v3/OS-FEDERATION/identity_providers/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	身份提供商的 ID。

请求

● Request Header参数说明

1				
参数	是否 必选	类型	说明	
Content- Type	是	String	该字段内容填为 "application/json;charset=utf8"。	
X-Auth- Token	是	String	具有Security Administrator权限token。	

● Request Body参数说明

参数	是否必 选	类型	说明
descriptio n	否	String	身份提供商的描述信息。
enabled	否	Boolean	身份提供商是否启用,true为启用,false 为停用,默认为false。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X PUT -d' {"identity_provider":{"description":"Stores ACME identities.", "enabled":true}}' https://10.185.190.118:31943/v3/0S-FEDERATION/identity_providers/ACME

响应

● Response Body参数说明

参数	是否必 选	类型	说明
id	是	String	身份提供商的ID。
description	是	String	身份提供商的描述信息。
enabled	是	Boolean	身份提供商是否启用。
remote_ids	是	Array	身份提供商的远程ID列表。
links	是	Object	身份提供商的资源链接,包含protocols, self。

● 响应样例

状态码

状态码	说明
201	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.1.4 更新身份提供商

功能介绍

该接口用于更新身份提供商信息。

URI

● URI格式

PATCH /v3/OS-FEDERATION/identity_providers/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	身份提供商的 ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	具有Security Administrator权限的token。

● Request Body参数说明

参数	是否必 选	类型	说明
descripti on	否	String	身份提供商的描述信息。
enabled	否	Boolea n	身份提供商是否启用,true为启用,false为停用,默认为false。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X PATCH -d' {"identity_provider":{"enabled":false}}' https://
10.185.190.118:31943/v3/OS-FEDERATION/identity_providers/ACME

响应

● Response Body参数说明

参数	t	是否必选	类型	说明
id		是	String	身份提供商的ID。

参数	是否必选	类型	说明
descriptio n	是	String	身份提供商的描述信息。
enabled	是	Boolean	身份提供商是否启用。
remote_id	是	Array	身份提供商的远程ID列表。
links	是	Object	身份提供商的资源链接,包含protocols, self。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
409	资源冲突。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.1.5 删除身份提供商

功能介绍

该接口用于删除身份提供商信息。

URI

● URI格式

DELETE /v3/OS-FEDERATION/identity_providers/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	身份提供商的 ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	具有Security Administrator权限的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X DELETE https://10.185.190.118:31943/v3/OS-FEDERATION/identity_providers/ACME

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.2 映射

4.9.2.1 查询映射列表

功能介绍

该接口用于查询映射列表信息。

URI

URI格式

GET /v3/OS-FEDERATION/mappings

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	查询映射列表时 所属租户的 token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.185.190.118:31943/v3/OS-FEDERATION/mappings

响应

● Response Body参数说明

参数	是否必选	类型	说明
mappings	是	Array	映射列表。
links	是	Object	映射的资源链 接,包含next, previous, self。

● 响应样例

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.2.2 查询映射

功能介绍

该接口用于查询映射信息。

URI

● URI格式 GET /v3/OS-FEDERATION/mappings/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	查询映射时所属 租户的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.185.190.118:31943/v3/OS-FEDERATION/mappings/ACME

响应

● Response Body参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。

参数	是否必选	类型	说明
rules	是	Object	将远端用户映射为本地用户的规则列表。 示例: "rules":[
			local中: ● user: 本地用户。
			● user: 本地用户。 ● group: 本地用户组。
			remote中:
			● "type": "UserName"表示IdP断言中的属性。
			● "type": "orgPersonType"表示IdP断言中 的属性。
links	是	Object	映射的资源链接,包含self。

● 响应样例

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.2.3 注册映射

功能介绍

该接口用于注册一个映射。

URI

● URI格式

PUT /v3/OS-FEDERATION/mappings/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	为已认证用户创建映射时所需的Security Administrator权限token。

● Request Body参数说明

Tules 是 Object 将远端用户映射为本地用户的规则列表。 示例:
● "type": "UserName"表示 IdP断言中的属性。

● 请求样例

```
curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token:$token" -X PUT -d' {"mapping": {"rules":[{"local":[{"user":{"name":"{0}"}}, {"group":{"name":"0cd5e9"}}], "remote":[{"type":"UserName"}, {"type":"orgPersonType", "not_any_of":["Contractor", "Guest"]}]}}' https:// 10. 185. 190. 118:31943/v3/OS-FEDERATION/mappings/ACME
```

响应

● Response Body参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。
rules	是	Object	将远端用户映射为本地用户的规则列表字符串。 示例: "rules": ["local": ["user": " "mame": "(0)" "remote": ["type": " "type: "orgPersonType" IdP 断言中的属性。 type: "orgPersonType" IdP 断言中的属性。
links	是	Object	映射的资源链接,包含 self。

● 响应样例

状态码

状态码	说明
201	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.2.4 更新映射

功能介绍

该接口用于更新映射信息。

URI

● URI格式

PATCH /v3/OS-FEDERATION/mappings/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	更新已认证用户名下的映射时所需的Security Administrator权限token。

● Request Body参数说明

参数	是否必选	类型	说明
rules	是	Object	将远端用户映射为本地用户的规则列表字符 串。 示例: "rules":[
			local中:
			● user 本地用户。
			● group 本地用户组。
			remote中:
			● "type": "UserName"表示IdP断言中的属性。
			● "type": "orgPersonType"表示IdP断言中的属性。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token: \$token" -X PATCH -d' {"mapping": {"rules": [{"local": [{"user": {"name": "{0}"}}, {"group": {"name": "0cd5e9"}}], "remote": [{"type": "UserName"}, {"type": "orgPersonType", "any one_of": ["Contractor", "SubContractor"]}]}]}' https:// 10. 185. 190. 118:31943/v3/OS-FEDERATION/mappings/ACME

响应

● Response Body参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。

参数	是否必选	类型	说明
rules	是	Object	将远端用户映射为本地用户的规则列表。 "rules": [
			● user 本地用户。
			● group 本地用户组。
			remote中:
			● type:"UserName" IdP断言中的属性。
			● type:"orgPersonType" IdP断言中的属性。
links	是	Object	映射的资源链接,包含self。

● 响应样例

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
409	资源冲突。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.2.5 删除映射

功能介绍

该接口用于删除映射信息。

URI

● URI格式 DELETE /v3/OS-FEDERATION/mappings/{id}

● 参数说明

参数	是否必选	类型	说明
id	是	String	映射的ID。

请求

● Request Header参数说明

参数	是否为必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	删除已认证用户 名下的映射时所 需的Security Administrator权限 token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X DELETE https://10.185.190.118:31943/v3/OS-FEDERATION/mappings/ACME

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.3 协议

4.9.3.1 查询协议列表

功能介绍

该接口用于查询协议列表信息。

URI

● URI格式

GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的 ID。

请求

● Request Header参数说明

参数	是否必 选	类型	说明
Content- Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth- Token	是	String	查询Protocol列表时所属租户的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.185.190.118:31943/v3/OS-FEDERATION/identity_providers/ACME/protocols/

响应

● Response Body参数说明

参数	是否必选	类型	说明
protocols	是	List of objects	Protocol的列表。
links	是	Object	Protocol的资源链接,包含next, previous,self。

● 响应样例

```
{
    "links": {
        "next": null,
        "previous": null,
        "self": "https://example.com/v3/OS-FEDERATION/identity_providers/ACME/protocols"
},
```

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.3.2 查询协议

功能介绍

该接口用于查询协议信息。

URI

URI格式
 GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的 ID。
protocol_id	是	String	Protocol的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content- Type	是	String	该字段内容填为 "application/json;charset=utf8"。
X-Auth- Token	是	String	查询Protocol时所属租户的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://10.185.190.118:31943/v3/0S-FEDERATION/identity_providers/ACME/protocols/saml

响应

● Response Body参数说明

参数	是否必 选	类型	说明
id	是	String	Protocol的ID。
mapping_i	是	String	Mapping的ID。
links	是	Object	Protocol的资源链接,包含identity_provider, self。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。

状态码	说明
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.3.3 注册协议

功能介绍

该接口用于注册一个协议。

URI

- URI格式
 PUT /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}
- 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的 ID。
protocol _id	是	String	Protocol的ID。当 前ID只支持 saml。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	为已认证用户创建Protocol时所需的SecurityAdministrator权限token。

● Request Body参数说明

参数	是否必选	类型	说明
mapping_id	是	String	Mapping的ID。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X PUT -d' {"protocol":{"mapping_id":"ACME"}}' https://
10.185.190.118:31943/v3/0S-FEDERATION/identity_providers/ACME/protocols/saml

响应

● Response Body参数说明

参数	是否必选	类型	说明
id	是	String	Protocol的ID。
mapping_id	是	String	Mapping的ID。
links	是	Object	Protocol的资源链接,包含 identity_provider , self。

● 响应样例

状态码

状态码	说明
201	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。

状态码	说明
503	服务不可用。

4.9.3.4 更新协议

功能介绍

该接口用于更新协议信息。

URI

● URI格式

PATCH /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的 ID。
protocol _id	是	String	Protocol的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	更新已认证用户的Protocol时所需的SecurityAdministrator权限token。

● Request Body参数说明

参数	是否必选	类型	说明
mapping_id	否	String	Mapping的ID。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\\$token" -X PATCH -d' \{"protocol": \{"mapping_id": "ACME"\}\}' https://
10.185.190.118:31943/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml

响应

● Response Body参数说明

参数	是否必选	类型	说明
id	是	String	Protocol的ID。
mapping_id	是	String	Mapping的ID。
links	是	Object	Protocol的资源链接,包含 identity_provider , self。

● 响应样例

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
409	资源冲突。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.3.5 删除协议

功能介绍

该接口用于删除协议信息。

URI

● URI格式

DELETE /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的 ID。
protocol _id	是	String	Protocol的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	删除已认证用户名下的 Protocol时所需的Security Administrator权限token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X DELETE https://10.185.190.118:31943/v3/OS-FEDERATION/identity_providers/ACME/protocols/saml

响应

无响应体。

状态码

状态码	说明
204	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。

状态码	说明
503	服务不可用。

4.9.4 Metadata

4.9.4.1 查询 Metadata 文件

功能介绍

该接口用于查询身份提供商导入到IAM中的Metadata文件内容。

URI

● URI格式

GET /v3-ext/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/metadata

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的ID。
protocol _id	是	String	Protocol的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	查询用户的Metadata文件内容时所需的Security Administrator权限 token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://lo.185.190.118:31943/v3-ext/OS-FEDERATION/identity_providers/ACME/protocols/saml/metadata

响应

● Response Body参数说明

参数	是否必选	类型	说明
id	是	String	Metadata的ID。
idp_id	是	String	Identity Provider的 ID。
entity_id	是	String	Metadata文件中的 的entityID字段。
protocol_id	是	String	Protocol的ID。
domain_id	是	String	用户所在domain 的ID。
xaccount_type	是	String	表示租户来源的 字段,默认为 空。
update_time	是	String	导入或更新 Metadata文件的时 间。
data	是	String	Metadata文件的内容。

● 响应样例

```
{
"id": "40c174f35ff94e31b8257ad4991bce8b",
"idp_id": "ACME",
"entity_id": "https://idp.test.com/idp/shibboleth",
"protocol_id": "saml",
"domain_id": "ed7a77d365304f458f7d0a7909c6d889",
"xaccount_type": "",
"update_time": "2016-10-26T09:26:23.000000",
"data": "$data"}
```

状态码

状态码	说明
200	获取成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
500	内部服务错误。

4.9.4.2 查询 Keystone 的 Metadata 文件

功能介绍

该接口用于查询keystone的Metadata文件。

URI

URI格式

GET /v3-ext/auth/OS-FEDERATION/SSO/metadata

请求

● Request Header参数说明

参数	是否必选	类型	说明
unsigned	否	Boolean	是否按SAML2.0规范,对元数据做 签名,默认false。

● 请求样例

GET /v3-ext/auth/OS-FEDERATION/SSO/metadata

响应

响应样例

```
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"</pre>
ID="43ebac773925f6849b196a3c803baba5" entityID="https://www.example.com">
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
\verb| (ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/> | (ds:CanonicalizationMethod Algorithm="http://www.
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference URI="#43ebac773925f6849b196a3c803baba5">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>yuQJc60I3xilt6X4c0EUBnVV2Vs=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>...</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"</pre>
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:KeyDescriptor use="encryption">
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data>
<ds:X509Certificate>...</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:NameIDFormat xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
urn:oasis:names:tc:SAML:2.0:nameid-format:transient
</md:NameIDFormat>
<md:AssertionConsumerService xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"</pre>
```

Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://www.example.com/v3-ext/auth/OS-FEDERATION/SSO/SAML2/POST" index="0" isDefault="true"/>

<md:AssertionConsumerService xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"

Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS" Location="https://www.example.com/v3-ext/auth/OS-FEDERATION/SSO/SAML2/ECP" index="1"/>

</md:SPSSODescriptor>

</md:EntityDescriptor>

状态码

状态码	说明
200	请求成功。
500	内部服务错误。
503	服务不可用。

4.9.4.3 导入 Metadata 文件

功能介绍

租户使用联邦认证功能时,需要先将Metadata文件导入IAM中,该接口用于导入租户的 Metadata文件。

URI

● URI格式

 $POST\ /v3-ext/OS-FEDERATION/identity_providers/\{idp_id\}/protocols/\{protocol_id\}/metadata$

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	已经注册的Identity Provider的ID。
protocol _id	是	String	已经注册的Protocol的 ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8"。
X-Auth-Token	是	String	更新用户的Metadata文件时 所需的Security Administrator权限token。

● Request Body参数说明

参数	是否必选	类型	说明
xaccount_type	是	String	该字段为标识租户来源字 段,默认为空。
metadata	是	String	该字段为用户IDP服务器的 Metadata文件的内容。
domain_id	是	String	用户所在domain的ID。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json; charset=utf8' -H "X-Auth-Token:\$token" -X POST -d '{"xaccount_type":"","domain_id":"ed7a77d365304f458f7d0a7909c6d889","metadata":"\$metadataContent"}' https://10.185.190.118:31943/v3-ext/OS-FEDERATION/identity_providers/ACME/protocols/saml/metadata

响应

响应样例

{ "message": "Import metadata successful"}

状态码

状态码	说明
201	导入成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
500	内部服务错误。

4.9.5 token

4.9.5.1 获取联邦认证的 unscoped token

功能介绍

该接口用于通过联邦认证方式获取unscope token。

URI

● URI格式 GET /v3/OS-FEDERATION/identity_providers/{idp_id}/protocols/{protocol_id}/auth

● 参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的ID。
protocol_id	是	String	Protocol的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
idp_id	是	String	Identity Provider的ID。
protocol_id	是	String	Protocol的ID。
Accept	否	string	通过增强客户端代理(ECP)获取 token时,该字段需取值如下: application/vnd.paos+xml
PAOS	否	string	增强客户端代理(ECP)获取 token时,该字段需取值如下: urn:oasis:names:tc:SAML: 2.0:profiles:SSO:ecp

∭说明

- 1. 该接口支持通过页面单点认证(WebSSO)和增强客户端代理(ECP)两种机制获取token,通过不同的请求header区分,具体使用请参考Request Header参数说明。
- 2. 不建议直接调用该接口,使用openstackclient获取token。
- 请求样例

 $\textit{GET /v3/OS-FEDERATION/} identity_providers/idptest/protocols/saml/auth$

响应

● Response Body参数说明

名称	参数	类型	说明
X-Subject-Token	header	string	签名后的unscoped token。
token	body	Object	联邦认证的unscoped token,包含methods和用户信息。

● 响应样例

```
"token": {
    "issued_at": "2017-05-23T06:54:51.763000Z",
    "expires_at": "2017-05-24T06:54:51.763000Z",
    "methods": [
        "mapped"
```

状态码

状态码	说明
200	请求成功,需进一步获取用户信息。
201	请求成功,返回token。
302	请求未携带Identity Provider用户信息时, 跳转到Identity Provider认证页面。
400	请求错误。
401	认证失败。
403	鉴权失败。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.5.2 获取联邦认证的 scoped token

功能介绍

该接口用于通过联邦认证方式获取scoped token。

URI

URI格式

POST /v3/auth/tokens

请求

● Request Header参数说明

参数名称	类型	是否必 选	说明
identity	Dict	是	鉴权信息 (用户密码)。
scope	Dict	否	Token所属范围信息。

● identity格式

参数名称	类型	是否 必选	说明
method s	List	是	token.
token	Dic t	是	联邦unscoped token。

● token格式

参数名 称	类型	是否 必选	说明
id	Stri ng	是	用于鉴权的联邦unscoped token id。

● scope格式

参数名 称	类型	是否 必选	说明
project	Dict	是	项目,与domain二选一。
domain	Dict	是	域,与project二选一。

● project格式

参数名 称	类型	是否必 选	说明
name	Strin g	是	项目名,与id二选一。
domain	Dict	是	项目所属域,使用name时必填。
id	Strin g	是	项目id,与name二选一。

domain格式

参数名称	类型	是否 必选	说明
name	Strin g	是	域名称,与id二选一。
id	Strin g	是	域id,与name二选一。

请求样例

```
POST /v3/auth/tokens
"auth": {
"identity": {
"methods": [
"token"
],
"token": {
"id": "--federated-token-id--"
},
"scope": {
    "domain": {
"id": "e31ac82d778b4d128cb6fed37fd72cdb"
```

∭说明

不建议直接调用该接口,使用openstackclient获取token。

响应

Response Body参数说明

参数名称	类型	是否必选	说明
methods	List	是	获取token时使用的鉴权方式。
roles	List	是	获取的token所代表的用户在项目或 域中的角色信息。
expires_at	String	是	token的失效时间。
project	Dict	是	token所属项目信息。
catalog	Dict	否	服务及终端节点信息。
extras	Dict	否	token额外信息。
user	Dict	是	token所属用户信息。
issued_at	String	是	token生成的时间。

响应样例

X-Subject-Token: MIIFwAYJKoZIhvcNAQcCoIIFsTCCBaOCAQExDTALBglghkgBZQMEAgEwggQOBgkqhkiG9w0BBwGgggP-BIID +3sidG9rZW4iOnsibWVOaG9kcyI6WyJ0b2tlbiJdLCJpc3N1ZWRfYXQiOiIyMDE3LTA1LTIzVDA2OjU00jEyLjUwODAwM FoilCJ1eHBpcmVzX2F0IjoiMjAxNyOwNSOyNFQwNjo1NDoxMi41MDgwMDBaIiwidXN1ciI6eyJpZCI6I1JNUVRndGpqU0 5HRGNLeTdvVW1JM0FaZzdHZ3NXRzBaIiwibmFtZSI6InN0b25laWRwMDEiLCJPUy1GRURFUkFUSU90Ijp7Im1kZW50aXR 5X3Byb3ZpZGVyIjp7ImlkIjoic3RvbmVpZHAwMSJ9LCJwcm90b2NvbCI6eyJpZCI6InNhbWwifSwiZ3JvdXBzIjpbeyJp ZCI6ImI0MDE40WUyNmVhNDRmOTU50Dc3NjIxYjRiMjk4ZGI1In1dfSwiZG9tYWluIjp7ImlkIjoiZTMxYWM4MmQ3NzhiN GQxMjhjYjZmZWQzN2ZkNzJjZGIiLCJuYW11Ijoic3RvbmUiLCJ4ZG9tYW1uX21kIjoieGRvbWFpbm1kMDA2OTU4MTQ5MD M5NDQ1NzEONDU5MzIzIiwieGRvbWFpb190eXB1IjoiVFNJIn19LCJkb21haW4iOnsiaWQiOiJ1MzFhYzgyZDc30GIOZDE y0GNiNmZ1ZDM3ZmQ3MmNkYiIsIm5hbWUi0iJzdG9uZSIsInhkb21haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG9tYW1uaWQwMDY5NTgxNDkwMzk01haW5faWQi0iJ4ZG1haW5faWQi0iJ4ZG1haW5faWQi0iJ4ZG1haW5faWQi0iJ4ZG1haW5faWQi0iJ4ZG1haW5faW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haWfaW0iJ4ZG1haW0iJNDU3MTQ0NTkzMjMiLCJ4ZG9tYW1uX3R5cGUi0iJUU0kifSwicm9sZXMi01t7Im1kIjoiZWF10DI2Njg0ZDc3NDYyNDgyZ DgxNThjMGZjN2IxNjEiLCJuYW11IjoidGVfYWRtaW4ifSx7ImlkIjoiMDA3YjczYjIyOWYxNGMzZDhlNzFmNWRjY2Y5Nj Y5YTYiLCJuYW11Ijoic2VjdV9hZG1pbiJ9LHsiaWQiOiI5M2JjNTc1M2UwZmMOZjAxYTZmZDY5ZjQ1YTE1YzEyNiIsIm5 hbWUiOiJOZV9hZ2VuY3kifSx7ImlkIjoiMCIsIm5hbWUiOiJvcF9nYXR1ZF9zdG9uZSJ9LHsiaWQiOiIwIiwibmFtZSI6 Im9wX2dhdGVkX3Rhc3NzZzEifSx7Im1kIjoiMCIsIm5hbWUi0iJvcF9nYXR1ZF90YXNzc2cyIn0seyJpZCI6IjAiLCJuY W11Ijoib3BfZ2F0ZWRfdGFzc3NnNCJ9LHsiaWQi0iIwIiwibmFtZSI6Im9wX2dhdGVkX3Rhc3NzZzUifSx7ImlkIjoiMC Is Im 5hb WU i O i JvcF 9n YXR 1ZF 90 YXN 2c 2c 2In 1d LC J jYXR hb G 9n I jpb XX 19M Y IB hTCC A YEC A QEwXD B XMQ sw CQYD VQQ G Ew JUR A CONTROL OF A CONTROLVUzEOMAwGA1UECAwFVW5zZXQxDjAMBgNVBAcMBVVuc2V0MQ4wDAYDVQQKDAVVbnN1dDEYMBYGA1UEAwwPd3d3LmV4YW1w bGUuY29tAgEBMAsGCWCGSAF1AwQCATANBgkqhkiG9w0BAQEFAASCAQBbiTxUJJ70SykOXspQwu5f8labMMjpM8clbe3PrPZNQhBtJNqG1joUH9QIWXJkQ54VHu9B0yWzO8enbn2qQaHu6IVzs4tAl034k25OCc YcBL241KJQtKDgJyu0Q1mnQXWCCcV9a5-3sQitvBYSINirYAh7UH-1Uh04q01nUp103UE0q6xhLpCy63DP7LgrfE8tIvkRxfj62-NgVffaEgxSC7iCZMc84MQYxdYWPXTrJk110UUh86JyzXf0EovsIWBGvC6g9FpPpUvT1pM+IK7yogFxmZwIshPLmDj5aqtaT6YxkMxMIY9G7kNC1jTUn1QJhqbIEIM-5z14f7m6w"token": { "domain": { "xdomain_type": "TSI", "id": "e31ac82d778b4d128cb6fed37fd72cdb", "xdomain_id": "xdomainid006958149039445714459323", "name": "exampledomain" "methods":["token"], "roles": ["id": "eae826684d77462482d8158c0fc7b161", "name": "te_admin" "id": "007b73b229f14c3d8e71f5dccf9669a6", "name": "secu_admin" "id": "93bc5753e0fc4f01a6fd69f45a15c126", "name": "te_agency" "id": "0", "name": "op_gated_stone" "id": "0", "name": "op_gated_tasssgl" "id": "0", "name": "op_gated_tasssg2" "id": "0", "name": "op_gated_tasssg4" "id": "0", "name": "op_gated_tasssg5" "id": "0", name": "op_gated_tasssg6"

```
"expires_at": "2017-05-24T06:54:12.508000Z",
"catalog": [
            "endpoints": [
                      "url": "https://sample.domain.com/v3",
                      "interface": "public",
"region": "*",
"region_id": "*",
"id": "f2a24165ecf14efeb5fcb2682ebc4cde"
           ],
"type": "identity",
"id": "90ded4a66ee14ecea72266ee2fdc2b0a",
"name": "keystone"
],
"user": {
      "OS-FEDERATION": {
           "identity_provider": {
    "id": "stoneidp01"
           "protocol": {
    "id": "saml"
            "groups": [
                {
                      "id": "b40189e26ea44f959877621b4b298db5"
           ]
     },
"domain": {
           "xdomain_type": "TSI",
           "id": "e31ac82d778b4d128cb6fed37fd72cdb",
           "xdomain_id": "xdomainid006958149039445714459323",
"name": "exampledomain"
      "id": "RMQTgtjjSNGDcKy7oUmI3AZg7GgsWGOZ",
"name": "exampleuser"
},
"issued_at": "2017-05-23T06:54:12.508000Z"
```

状态码

状态码	说明
201	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
500	内部服务错误。
503	服务不可用。

4.9.6 租户

4.9.6.1 查询联邦用户可以访问的租户列表

功能介绍

该接口用于查询联邦用户可以访问的租户列表。

URI

URI格式

GET /v3/OS-FEDERATION/domains

请求

Request Header参数说明

参数	是否必选	类型	说明
X-Auth-Token	是	String	Unscope token, 请参考 获取联邦 认证的unscoped token。

∭说明

推荐使用查询用户可以访问的租户列表,该接口可以返回相同的响应格式。

● 请求样例

GET /v3/OS-FEDERATION/domains

响应

● Response Body参数说明

参数	是否必选	类型	说明
domains	是	array	租户列表。
links	是	Object	资源访问链接。

● 响应样例

```
"links": {
    "self": "https://sample.domain.com/v3/OS-FEDERATION/domains",
    "previous": null,
    "next": null
}
```

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.7 项目

4.9.7.1 查询联邦用户可以访问的项目列表

功能介绍

该接口用于查询联邦用户可以访问的项目列表,需要项目列表来获取联邦scoped token。

URI

URI格式

GET /v3/OS-FEDERATION/projects

请求

● Request Header参数说明

参数	是否必 选	类型	说明
X-Auth-	是	Stri	Unscoped token,请参考 获取联邦认证的unscoped
Token		ng	token获取token。

□说明

推荐使用查询用户可以访问的项目列表,该接口返回相同的响应格式。

● 请求样例

GET /v3/OS-FEDERATION/projects

响应

● Response Body参数说明

参数	是否必选	类型	说明
projects	是	array	项目列表。
links	是	Object	资源访问链接。

● 响应样例

```
"links": {
    "self": "https://sample.domain.com/v3/OS-FEDERATION/projects",
  "previous": null,
  "next": null
"projects": [
     "is_domain": false,
     "description": "
     "links": {
    "self": "https://sample.domain.com/v3/projects/05cf683c351e43518618d9fa96a5efa9"
     "enabled": true,
"id": "05cf683c351e43518618d9fa96a5efa9",
     "parent_id": "e31ac82d778b4d128cb6fed37fd72cdb",
"domain_id": "e31ac82d778b4d128cb6fed37fd72cdb",
     "name": "region_name"
     "is_domain": false,
     "description": "",
     "links": {
    "self": "https://sample.domain.com/v3/projects/32b56f108f87418e8219317beb0fff3c"
     "enabled": true,
     "id": "32b56f108f87418e8219317beb0fff3c",
     "parent_id": "e31ac82d778b4d128cb6fed37fd72cdb", "domain_id": "e31ac82d778b4d128cb6fed37fd72cdb",
     "name": "MOS"
]
```

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。

状态码	说明
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.9.8 断言

4.9.8.1 WebSSO 断言处理接口

功能介绍

当IAM服务担任SAML2.0规范中的SP(Service Provider,即服务提供者)角色与IDP(Identity Provider,即身份提供者)进行单点登录时,该接口用于接收AuthnRequest请求返回的响应,即IDP发往SP的断言请求。

关于SAML2.0规范请参阅链接: https://en.wikipedia.org/wiki/SAML_2.0

□说明

不建议直接调用,请使用openstackclient获取token。

URI

URI格式

POST /v3-ext/auth/OS-FEDERATION/SSO/SAML2/POST

4.9.8.2 ECP 断言处理接口

功能介绍

当IAM服务担任SAML2.0规范中的SP(Service Provider,即服务提供者)角色与IDP(Identity Provider,即身份提供者)进行单点登录时,该接口用于接收AuthnRequest请求返回的响应,即客户端代理发往SP的断言请求。

关于SAML2.0规范请参阅链接: https://en.wikipedia.org/wiki/SAML_2.0

∭说明

不建议直接调用,请使用openstackclient获取token。

URI

URI格式

POST /v3-ext/auth/OS-FEDERATION/SSO/SAML2/ECP

4.10 版本信息

4.10.1 查询 Keystone API 的版本信息

功能介绍

该接口用于获取Keystone API的版本信息。

URI

URI格式

GET /

请求

请求样例

curl -i -k -X GET https://www.example.com/

响应

● 响应参数说明

参数	是否必选	类型	描述
versions	是	Object	Keystone API的版本信息。
values	是	Array	Keystone API的版本列表信息。

● values格式说明

参数	是否必选	类型	说明
status	是	String	版本状态。
updated	是	String	版本最后更新时间。
media-types	是	Array	版本支持的消息格 式。
id	是	String	版本号,如v3.0。
links	是	Array	版本的链接。

● 响应样例 (响应成功)

状态码

状态码	说明
300	请求成功。
400	请求错误。
404	找不到资源。
503	服务不可用。

4.10.2 查询 Keystone API 的 3.0 版本信息

功能介绍

该接口用于获取Keystone API的3.0版本的信息。

URI

URI格式

GET/v3

请求

请求样例

curl -i -k -X GET https://www.example.com/v3

响应

● 响应参数说明

参数	是否必 选	类型	描述
version	是	Object	Keystone API的版本信息。

● version格式说明

参数	是否必选	类型	说明
status	是	String	版本状态。
updated	是	String	版本最后更新时间。
media-types	是	Array	版本支持的消息格 式。
id	是	String	版本号,如v3.0。
links	是	Array	版本的链接。

● 响应样例(响应成功)

状态码

状态码	说明
200	请求成功。
400	请求错误。
404	找不到资源。
503	服务不可用。

4.11 服务和终端节点

4.11.1 查询服务列表

功能介绍

该接口用于查询服务列表。

URI

- URI格式 GET /v3/services {?type}
- 参数说明

参数	是否必选	类型	说明
type	否	String	服务类型,类型 的值可以是 compute,ec2, identity, image, network, 或 volume。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/services?type=compute

响应

● Reponse Body参数说明

参数	是否必选	类型	说明
links	是	Dict	服务的链接,包 含 next,previous,self 。
services	是	List	服务列表。

● services格式说明

参数	是否必选	类型	说明
description	否	String	服务描述。
enabled	是	Boolean	服务是否可用。

参数	是否必选	类型	说明
id	是	String	服务id。
name	否	String	服务名。
type	是	String	服务类型。
links	是	Dict	服务的链接。

响应样例 (响应成功)

```
"services": [
                "name": "compute5",
"links": {
                    "self": "https://iamcore_links.com/v3/services/
053d21d488d1463c818132d9d08fb617"
               },
  "enabled": true,
  "type": "compute",
  "id": "053d21d488d1463c818132d9d08fb617",
  "lambertion": "Compute service 5"
                "name": "compute3",
"links": {
                     "self": "https://iamcore_links.com/v3/services/
c2474183dca7453bbd73123a0b78feae"
               },
"enabled": true,
"type": "compute",
"id": "c2474183dca7453bbd73123a0b78feae",
"description": "Compute service 3"
                "name": "compute2",
"links": {
                      "self": "https://iamcore_links.com/v3/services/
c7166694ebdd4616bd927737f7b12ca2"
                },
"enabled": true,
                "type": "compute",
"id": "c7166694ebdd4616bd927737f7b12ca2",
                "description": "Compute service 2"
    ],
"links": {
    "-plf"
          "self": "https://iamcore_links.com/v3/services?type=compute",
          "previous": null,
          "next": null
```

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。

状态码	说明
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.11.2 查询服务详情

功能介绍

该接口用于查询服务详情。

URI

● URI格式 GET /v3/services/{service_id}

● 参数说明

参数	是否必选	类型	说明
service_id	否	String	指定服务的ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/services/5a4ed456d228428c800ed2b67b4363a7

响应

● 响应样例(响应成功)

```
"service": {
```

```
"enabled": true,
    "type": "compute",
    "name": "nova",
    "links": {
        "self": "10.10.10.10/v3/services/5a4ed456d228428c800ed2b67b4363a7"
    },
    "id": "5a4ed456d228428c800ed2b67b4363a7"
}
```

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.11.3 查询终端节点列表

功能介绍

该接口用于查询终端节点列表,提供服务访问入口。

URI

● URI格式

GET /v3/endpoints {? interface, service_id}

● 参数说明

参数	是否必选	类型	说明
interface	否	String	过滤终端节点平面,类型取值可以是public,internal或admin。
service_id	否	String	过滤服务id。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/endpoints?
interface=public&service_id=43cbe5e77aaf4665bbb962062dc1fc9d

响应

● Response Body参数说明

参数	是否必选	类型	说明
links	是	dict	终端节点的链接,包含next, previous, self。
endpoints	是	list	终端节点列表。

● endpoints格式说明

参数	是否必选	类型	说明
id	是	String	终端节点id。
url	是	String	终端节点的地 址。
region	是	String	终端节点的区 域。
region_id	是	String	终端节点的区域 id。
enabled	是	Boolean	终端节点是否可 用。
interface	是	String	终端节点的平 面。
service_id	是	String	终端节点所属服 务的id。
links	是	dict	终端节点的链 接。

● 响应样例(请求成功)

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.11.4 查询终端节点详情

功能介绍

该接口用于查询终端节点详情。

URI

- URI格式 GET /v3/endpoints/{endpoint_id}
- 参数说明

参数	是否必选	类型	说明
endpoint_id	是	String	指定的终端节点 ID。

请求

● Request Header参数说明

参数	是否必选	类型	说明
Content-Type	是	String	该字段内容填为 "application/ json;charset=utf8 "。
X-Auth-Token	是	String	已认证的token。

● 请求样例

curl -i -k -H 'Accept:application/json' -H 'Content-Type:application/json;charset=utf8' -H "X-Auth-Token:\$token" -X GET https://172.30.48.86:31943/v3/endpoints/62ea3602f8ee42b1825956473f5295a8

响应

● 响应样例(请求成功)

状态码

状态码	说明
200	请求成功。
400	请求错误。
401	认证失败。
403	鉴权失败。
404	找不到资源。
405	不允许的方法。

状态码	说明
413	请求体过大。
500	内部服务错误。
503	服务不可用。

4.12 相关参数信息获取

4.12.1 获取用户名称、项目名称、项目 ID

登录管理控制台,单击右上方登录的帐号,选择"我的凭证"。进入"我的凭证"界面,可以查看用户名称、项目名称、项目 \mathbf{ID} 。



A 文档修订记录

表 A-1 文档修订记录

发布时间	修改记录
2017-10-27	第十五次正式发布。
	本次变更说明如下:
	新增以下章节:
	● 查询指定项目信息及状态
	● 创建委托
	● 获取指定委托的详细信息
	● 修改委托
	● 删除委托
	● 通过项目给委托赋予权限
	● 检查委托在项目上是否有指定权限
	● 查询委托在项目上具有的权限列表
	● 通过项目删除委托的权限
	● 通过租户给委托赋予权限
	● 检查委托在租户上是否有指定权限
	● 查询委托在租户上具有的权限列表
	● 通过租户删除委托的权限
	● 通过assume_role方式获取用户token
2017-09-24	第十四次正式发布。
	本次变更说明如下:
	● 修改Token认证章节中获取token的body体中的scope中的 project为domain。
	● 删除 获取项目ID 章节中描述tenant_id的部分。
	● 删除 查询终端节点列表 章节中响应url中的tenant_id。
	● 在 项目 中增加不允许在MOS下创建project的注意项。

发布时间	修改记录
2017-08-28	第十三次正式发布。
	本次变更说明如下:
	新增 设置指定项目的状态 章节。
2017-07-27	第十二次正式发布。
	本次变更说明如下:
	新增以下章节:
	● 査询区域列表
	● 査询区域详情
	● 查询服务详情
	● 查询终端节点详情
	● 创建项目
	● 修改项目
	● 查询指定项目的信息
2017-06-28	第十一次正式发布。
	本次变更说明如下:
	增加 通过assume_role方式获取用户token 章节。
2017-06-09	第十次正式发布。
	本次变更说明如下:
	变更以下章节:
	● 查询用户所属用户组
	● 査询用户组
	● 査询用户组详情
	● 查询指定条件下的项目信息
	● 查询用户的项目列表
	● 查询用户可以访问的项目列表
	● 查询权限列表
	● 查询权限的详细信息
	● 查询租户中用户组的权限
	● 查询项目对应的用户组的权限
	● 为项目对应的用户组授权
	● 获取用户Token
	● 查询身份提供商列表
	● 查询身份提供商
	● 更新身份提供商
	● 查询服务列表
	● 获取用户名称、项目名称、项目ID

发布时间	修改记录
2017-4-27	第九次正式发布。 本次变更说明如下: ● 增加以下章节: - 删除用户 - 删除用户组中用户 - 查询用户组中用户 - 查询用户组 - 查询用户组 - 查询用户组 - 题新用户组 - 更新用户组 - 更新用户组 - 则除用户组 - 更新用户组 - 查询权限的详细信息 - 查询权限的详细信息 - 查询和户中用户组的权限 - 在户的用户组授权 - 为项目对应的用户组授权 - 为项目对应的用户组授权 - 删除项目对应的用户组的权限 - 删除项目对应的用户组的权限 - 删除项目对应的用户组的权限 - 删除项目对应的用户组是否包含权限 - 查询项目对应的用户组是否包含权限 - 查询项目对应的用户组是否包含权限 - 按照接口分类修改目录结构。
2017-03-28	第八次正式发布。 本次变更说明如下: 增加 查询Keystone API的3.0版本信息
2017-01-20	第七次正式发布。增加以下章节: ① 查询用户可以访问的租户列表 ② 查询用户可以访问的项目列表 ② 查询Keystone的Metadata文件 ③ 获取联邦认证的unscoped token ④ 获取联邦认证的scoped token ④ 查询联邦用户可以访问的项目列表 ④ 查询联邦用户可以访问的租户列表 ④ WebSSO断言处理接口 ⑥ ECP断言处理接口

发布时间	修改记录
2016-12-30	第六次正式发布。
	本次变更说明如下:
	● GET /v3/projects接口的响应增加page字段(查询页面的数据)。
	● GET /v3/projects接口的响应增加per_page字段(页面数据个数)。
2016-10-29	第五次正式发布。
	本次变更说明如下:
	增加以下章节:
	● 查询身份提供商列表
	● 查询身份提供商
	● 注册身份提供商
	● 更新身份提供商
	● 删除身份提供商
	● 查询映射列表
	● 査询映射
	● 注册映射
	● 更新映射
	● 删除映射
	● 査询协议列表
	● 査询协议
	● 注册协议
	● 更新协议
	● 删除协议
	● 导入Metadata文件
	● 查询Metadata文件
2016-09-30	第四次正式发布。
	本次变更说明如下:
	增加以下章节:
	● 査询用户列表
	● 査询用户详情
	● 查询用户所属用户组
	● 创建用户
	● 修改密码
	● 修改用户信息
	● 删除用户
	● 查询项目对应的用户组的权限

发布时间	修改记录
2016-08-25	第三次正式发布。 本次变更说明如下: 增加token中各字段的含义。
2016-06-30	第二次正式发布。 本次变更说明如下: ● GET /v3/services接口的响应增加links字段(服务的链接)。 ● GET /v3/endpoints接口的响应增加links字段(终端地址的链接)。
2016-03-14	第一次正式发布。