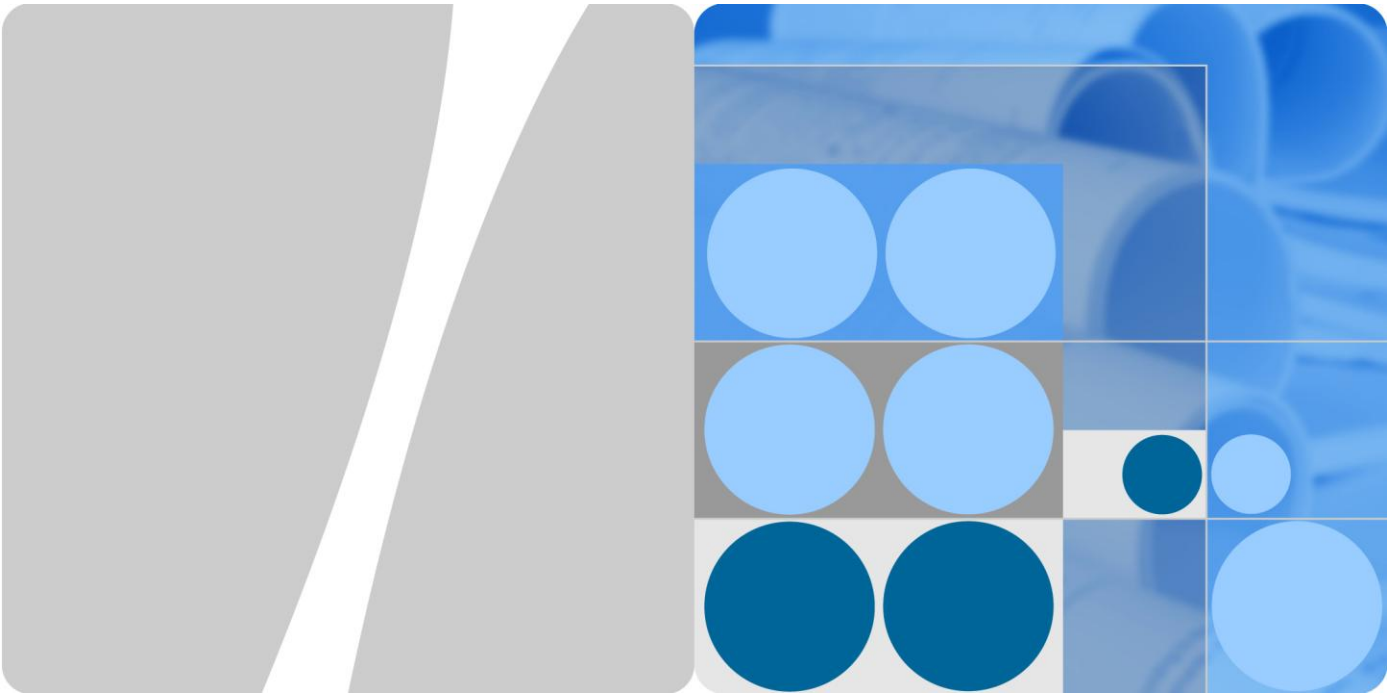


华为商业 WIFI eSight 接口文档

华为商业 WIFI eSight 接口文档

文档版本 01

发布日期 2016-06-03

华为技术有限公司

商标声明


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

前言

概述

介绍 Open API 的基础知识和相关配置，帮您快速掌握 Open API 相关操作。

读者对象

本文档主要适用于以下工程师：

- eSight 系统管理员
- 上层网管系统管理员

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
	用于警示紧急的危险情形，若不避免，将会导致人员死亡或严重的人身伤害。
	用于警示潜在的危险情形，若不避免，可能会导致人员死亡或严重的人身伤害。
	用于警示潜在的危险情形，若不避免，可能会导致中度或轻微的人身伤害。
	用于传递设备或环境安全警示信息，若不避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “注意”不涉及人身伤害。
	用于突出重要/关键信息、最佳实践和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

修改记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 01 (2016-05-30)

第一次发布版本。

目 录

eSight V300R006C00 Open API 用户指南

1 入门指南

本节主要介绍 Open API 的基础知识，帮您快速入门。

1.1 快速入门

什么是 Open API

Open API 是 eSight 提供的基于 REST（Representational State Transfer）标准的开放接口。第三方开发者能使用 Open API 接口授权访问 eSight 开放的资源，例如：安全管理、网元接入管理、告警管理等。

Open API 的使用场景

随着互联网发展的多元化，更多的技术人员投入到网络应用的开发上，及网络开发门槛降低化，让更多的技术人员可以更容易的发展多家互联网可使用的 API（Application Programming Interface，应用编程接口），更多的企业网站开发了 API 加大了技术人员在这方面投入的精力，未来几年将是 API 飞速发展的几年，让更多企业网站内容更丰富，与访问者更加互动。

Open API 如其名，是一种开放式的认证方式，通过非对称加密技术完成用户的认证，其返回相应的 OpenID 后，再实现资源的获取，实现不同平台、企业、实体之间的数据共享机制。

Open API 是服务型网站常见的一种应用，网站的服务商将自己的网站服务封装成一系列 API 开放出去，供第三方开发者使用，这种行为就叫做开放网站的 API，所开放的 API 就被称作 Open API（开放 API）。 Open API 的出现一方面是信息联合的需要，另一方面是来自利益驱动。

提供 Open API 调用方式目前已经流行于各大互联网公司，如阿里、谷歌等公司，都提供一套供第三方开发者调用的接口，通过此接口认证后，即可调用其数据单元内的海量资源。数据是有限的，创意是无限的，用海量的数据来完成无限的创意。

Open API 的特点

1. 提供标准的集成方案，有利于第三方系统快速集成。
2. 接口需要授权使用，访问安全，只支持 HTTPS 访问。
3. 采用 JSON（JavaScript Object Notation）数据格式进行数据交互，格式简单，易于读写，相比 XML 占用的网络流量较小。

什么是 REST

REST（Representational State Transfer）是一种针对网络应用的设计和开发方式，可以降低开发的复杂性，提高系统的可伸缩性。

REST 以资源为核心，资源由 URI（Uniform Resource Identifier）唯一标识，例如/rest/openapi/eam/meservice。

REST 采用 4 种标准操作访问资源：POST、GET、PUT、DELETE。

- POST 通常表示创建资源。
- GET 通常表示查询资源的信息。
- PUT 通常表示更新资源的信息。
- DELETE 通常表示删除资源。

eSight 的各种对外服务以 URI 的形式向用户提供，用户通过 URI 来获取 eSight 的资源，进而获取平台相应的服务，以实现用户的需求。

什么是 JSON

JSON 是一种轻量级的数据格式。

JSON 有两种结构：

- 对象：以{}括起来的内容，形如{key1: value1, key2: value2, ...}。可以通过 Object.key 形式访问数据。
- 数组：以[]括起来的内容，形如[element1, element2, ...]。可以通过 Object[index]形式访问数据，index 从 0 开始。

可见，相对于 XML，JSON 字符串更简短，更节省网络流量。

下面的示例代码展示如何访问 JSON 字符串的数据。

```
<script type="text/javascript">
  var resultInfo = {"data":{"total":7,"key_2":2,"key_1":5},
                "code":0,
                "description":"Success to get alarmcount." };
  //Access the value of key_1.
  var key_1 = resultInfo.data.key_1;
</script>
```

什么是 HTTP

超文本传输协议（英文：HyperText Transfer Protocol，缩写：HTTP）是互联网上应用最为广泛的一种网络协议。

设计 HTTP 最初的目的是为了提供一种发布和接收 HTML 页面的方法。

通过 HTTP 或者 HTTPS 协议请求的资源由统一资源标识符（Uniform Resource Identifiers，URI）来标识。

HTTPS（Hypertext Transfer Protocol over Secure Socket Layer），是以安全为目标的 HTTP 通道，简单讲是 HTTP 的安全版。HTTPS 的安全基础是 SSL，因此传输数据通过 SSL 加密，以确保安全性。

HTTP 请求信息（Request Message）

发出的请求信息包括以下几个：

- 请求行。例如 GET /images/logo.gif HTTP/1.1，表示从/images 目录下请求 logo.gif 这个文件。
- 请求头。例如 Accept-Language: en。
- 空行。
- 其他消息体。

HTTP 请求方法

HTTP/1.1 协议中共定义了八种方法（也叫“动作”）来以不同方式操作指定的资源：

- OPTIONS：这个方法可使服务器传回该资源所支持的所有 HTTP 请求方法。用'*'来代替资源名称，向 Web 服务器发送 OPTIONS 请求，可以测试服务器功能是否正常运作。
- HEAD：与 GET 方法一样，都是向服务器发出指定资源的请求。只不过服务器不传回资源的本文部份。它的好处在于，使用这个方法可以在不必传输全部内容的情况下，就可以获取其中“关于该资源的信息”（元信息或称元数据）。
- GET：向指定的资源发出“显示”请求。使用 GET 方法应该只用在读取数据，而不应当被用于产生“副作用”的操作中，例如在 Web Application 中。
- POST：向指定资源提交数据，请求服务器进行处理（例如提交表单或者上传文件）。数据被包含在请求本文中。这个请求可能会创建新的资源或修改现有资源，或二者皆有。
- PUT：向指定资源位置上传其最新内容。
- DELETE：请求服务器删除 Request-URI 所标识的资源。
- TRACE：回显服务器收到的请求，主要用于测试或诊断。
- CONNECT：HTTP/1.1 协议中预留给能够将连接改为管道方式的代理服务器。通常用于 SSL 加密服务器的链接（经由非加密的 HTTP 代理服务器）。

方法名称是区分大小写的。当某个请求所针对的资源不支持对应的请求方法的时候，服务器应当返回状态码 405（Method Not Allowed），当服务器不认识或者不支持对应的请求方法的时候，应当返回状态码 501（Not Implemented）。

HTTP 状态码

所有 HTTP 响应的第一行都是状态行，依次是当前 HTTP 版本号，3 位数字组成的状态代码，以及描述状态的短语，彼此由空格分隔。

状态代码的第一个数字代表当前响应的类型：

- 1xx 消息——请求已被服务器接收，继续处理
- 2xx 成功——请求已成功被服务器接收、理解、并接受
- 3xx 重定向——需要后续操作才能完成这一请求
- 4xx 请求错误——请求含有词法错误或者无法被执行
- 5xx 服务器错误——服务器在处理某个正确请求时发生错误

HTTP 请求样例

下面是一个 HTTP 客户端与服务器之间会话的例子，运行于 www.google.com，端口 80。

客户端请求：

```
GET / HTTP/1.1
Host:www.google.com
```

（第一行指定方法、资源路径、协议版本；第二行是在 1.1 版本必带的一个 header，作用为指定主机。）

服务器应答：

```
HTTP/1.1 200 OK
Content-Length: 3059
Server: GWS/2.0
Date: Sat, 11 Jan 2003 02:44:04 GMT
Content-Type: text/html
Cache-control: private
Set-Cookie: PREF=ID=73d4aef52e57bae9:TM=1042253044:LM=1042253044:S=SMCc HRPCQiqy
X9j; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.com
Connection: keep-alive
```

1.2 架构概述

eSight 为第三方开发者提供了一套 Open API 接口，第三方开发者可以使用 HttpClient 来调用 Open API 接口。

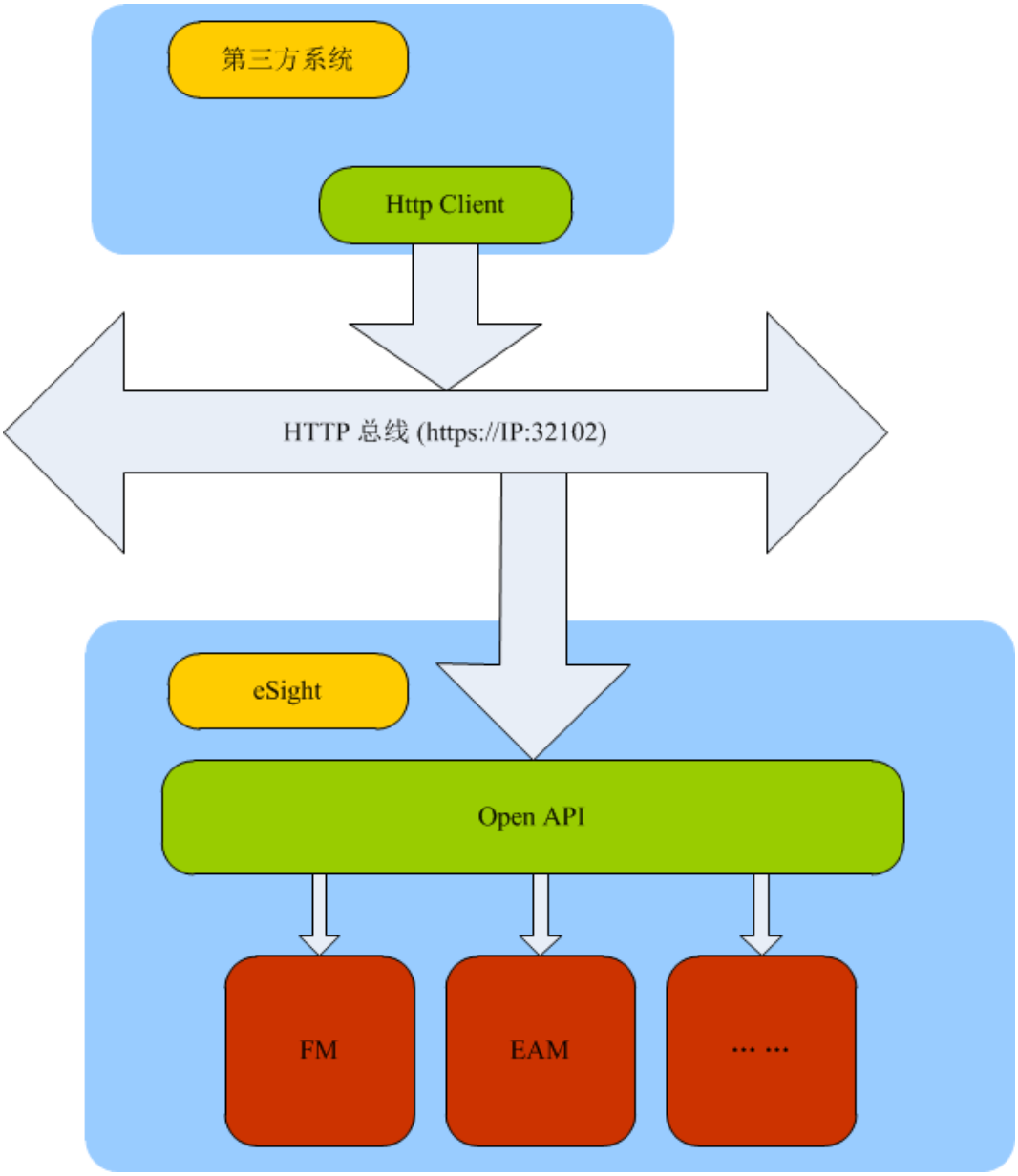
HttpClient 是一种支持 HTTP 协议的客户端编程工具包，它实现了 POST、GET、PUT、DELETE 等方法。由于 eSight 资源必须通过 HTTPS 访问，所以用户使用的 HttpClient 必须支持 HTTPS 协议。

Apache Jakarta Common 项目开发的 HttpClient 是 Java 版本的常用 HttpClient，详情请参考 http://hc.apache.org/httpcomponents-client-ga/index.html，开发示例详见 3.1 开发一个基于 HttpClient 的应用。用户也可以使用其他语言的 HttpClient 来调用 Open API 接口。

Open API 架构如图 1-1 所示。

eSight V300R006C00 Open API 用户指南

图1-1 Open API 架构图

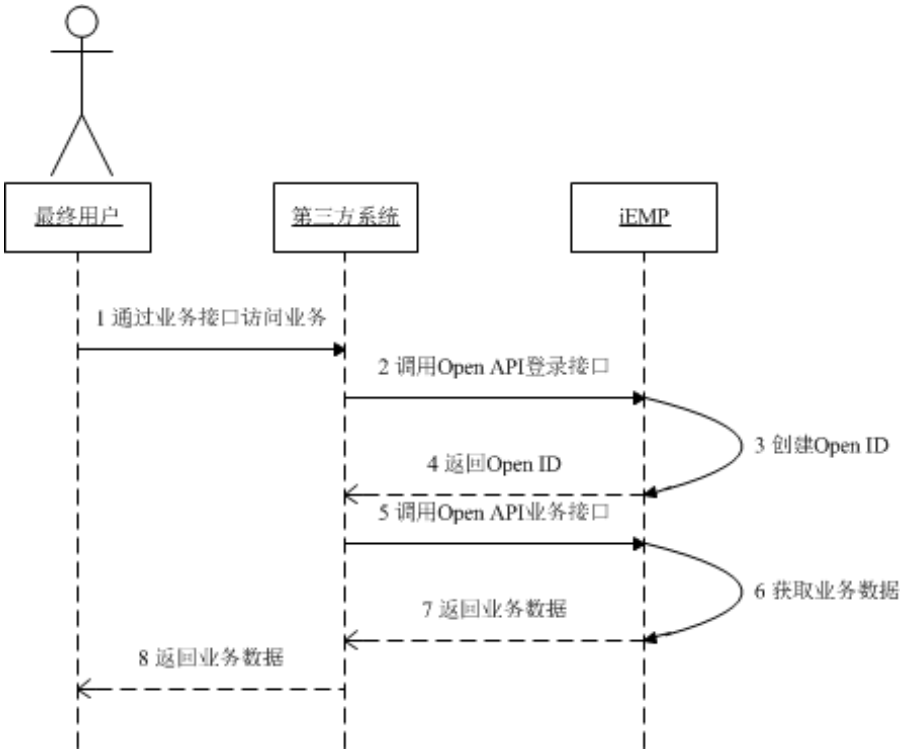


其中，FM（Fault Management）是故障管理模块，EAM（Element Access Management）是网元接入管理模块。

Open API 接口的调用流程如图 1-2 所示。

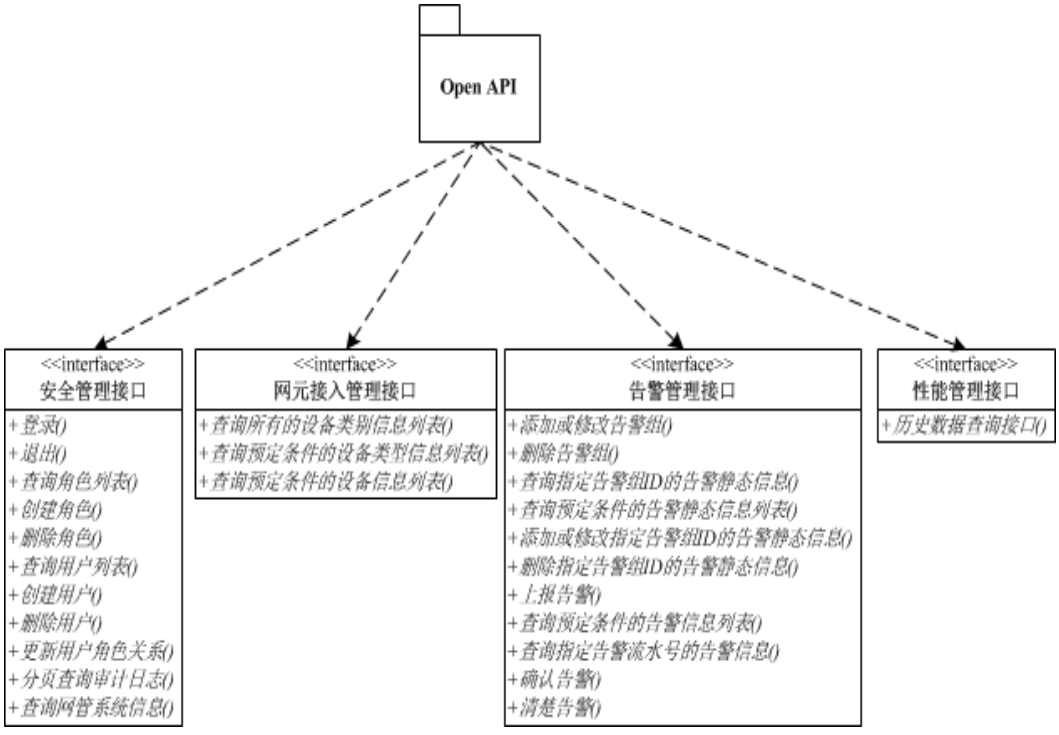
eSight V300R006C00 Open API 用户指南

图1-2 调用流程



Open API 提供的接口如图 1-3 所示。

图1-3 Open API 接口



eSight V300R006C00 Open API 用户指南

2 前期准备

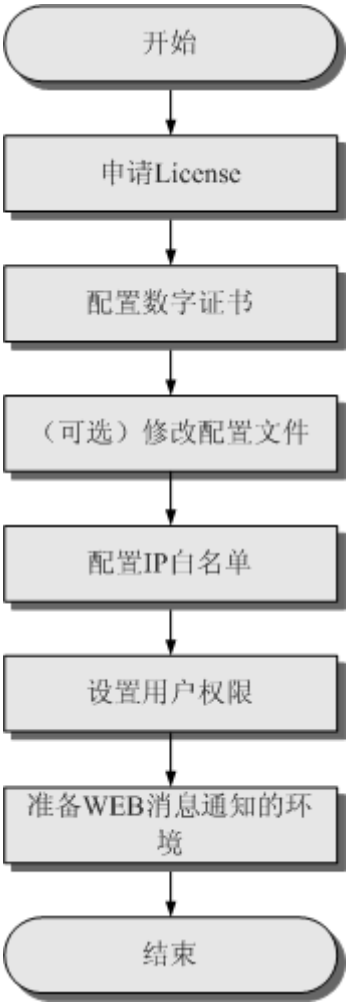
本节主要介绍 Open API 的前期准备。

2.1 基于 openid 认证

请参考图 2-1 所示做好基于 openid 认证的前期准备。

eSight V300R006C00 Open API 用户指南

图2-1 前期准备



申请 License

网管用户需要提前向华为公司申购 License，否则代理商无法调用 Open API。

用户申请的 License 控制项为“SNMP 告警北向功能&OpenSDK”（编号为 LNSHFSDK031）。

申请 License 的详细流程请参考《License 申请指导书》。

查看当前 License 是否有“SNMP 告警北向功能&OpenSDK”控制项的方法如下：

1. 在主菜单中选择“系统 > 系统管理 > License 管理”。
2. 在“License 管理”页面中，可以查看 License 基本信息、资源控制信息、功能控制信息和失效码信息。

配置数字证书

调用 Open API 是采用 HTTPS 方式，所以在建立 HTTPS 请求连接时需要证书。证书的默认路径是：<安装目录>/AppBase/etc/certificate/JettyServerKeyStore。

注意

- eSight 不提供证书，需要企业用户自己生成证书放至默认路径下。
- 请确保生成的证书密码与 ros.xml 中的配置项 keystorePass 一致；生成的证书存放路径与 ros.xml 中的配置项 keystorefilePath 一致。配置方法可参见表 2-1。

修改配置文件

网管管理员需要参照下表设置“<安装目录>/AppBase/etc/oms.ros/ros.xml”文件中的参数。

表2-1 ros.xml 配置项说明

参数	如何理解……	必选/可选	如何设置……	生效方式
webservers/rosOpenAPIROA/threadpool.thread.max	线程池最大线程数。	可选	**数据类型：** String **参数值：** 数字字符串 **缺省值：** 100	重启生效
webservers/rosOpenAPIROA/threadpool.queue.max	线程池队列最大值。	可选	**数据类型：** String **参数值：** 数字字符串 **缺省值：** 2000	重启生效
webservers/rosOpenAPIROA/roa.server.ip.white.list.enable	是否开启 IP 白名单。	可选	**数据类型：** 布尔型 **取值范围：** false 或 true **缺省值：** true	重启生效
webservers/rosOpenAPIROA/ip.white.type	白名单标识。	可选	**数据类型：** String **取值范围：** 字符串 **缺省值：** jetty.openapi	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ip	Open API 接口调用时的服务器 IP 地址。 说明 在实际使用中，需要将该 IP 地址替换为网管的 IP 地址（不能为 127.0.0.1	必选	**数据类型：** String **参数值：** 数字字符串表示的 IP 地址	重启生效

eSight V300R006C00 Open API 用户指南

参数	如何理解……	必选/可选	如何设置……	生效方式
	或 0.0.0.0），第三方才能调用 Open API 接口。否则，只有本机才能调用 Open API 接口。		**缺省值:**127.0.0.1	
webservers/rosOpenAPIROA/connectors/openapiROAConnector/port	调用 Open API 时的端口号。	必选	**数据类型：**String **参数值：**数字字符串表示的端口号 **缺省值：**32102	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.protocol	openapi 接口调用时的协议版本。 说明 • 该配置项默认不提供（此时默认支持 TLSv1.1 和 TLSv1.2），由用户自行配置。 • 由于 jre1.7 不完全支持 TLSv1.2，若使用 TLSv1.2 协议，推荐使用 jre1.8。	可选	**数据类型：**String **参数值：**支持使用的 https 协议版本 **缺省值：**无	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.keystore.path	HTTPS 连接时证书的存放路径。	必选	**数据类型：**String **取值范围：**路径字符串 **缺省值：**etc/certificate/JettyServerKeyStore	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.keystore.password	HTTPS 连接时证书的密码。 说明 该密码是使用 encrypt 生成的密文密码。 • 在 Windows 操作系统中，需要使用“<安装目录>/AppBase/tools/bmetool/encrypt/encrypt.bat”工具来加密存储密码，命令格式为 encrypt.bat 0。 • 在 Linux 操作系统中，需要使用“<安装目	必选	**数据类型：**String **参数值：**字符串表示的证书密码 **缺省值：**9d7961bc8af54d05ce509e03b13ffce3abc75873 73e7719b62555fd5aff9908d	重启生效

参数	如何理解……	必选/可选	如何设置……	生效方式
	录 >/AppBase/tools/bmetool/encrypt/encrypt.sh”工具来加密存储密码，命令格式为./encrypt.sh 0。			

配置举例：

```
<?xml version="1.0" encoding="UTF-8"?>
<webservers>
  <webserver name="rosOpenAPIROA">
    <property name="threadpool.thread.max" value="100" />
    <property name="threadpool.queue.max" value="2000" />
    <property name="roa.server.ip.white.list.enable" value="true" />
    <property name="ip.white.type" value="jetty.openapi" />
    <connectors>
       <connector name="openapiROAConnector" type="https">
          <property name="ip" value="127.0.0.1" />
          <property name="port" value="32102" />
          <property name="ssl.keystore.path"
value="etc/certificate/JettyServerKeyStore" />
          <property name="ssl.keystore.password"
value="9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555fd5aff9908d" />
        </connector>
     </connectors>
   </webserver>
</webservers>
```

配置 IP 白名单

网管管理员需要参照下表配置“<安装目录>/AppBase/etc/iemp.framework/ip.white.list/ip-white-list-openapi.properties”文件中的白名单。

说明
双机环境时，请网管管理员登录到主机路径“<安装目录>/eSightSync/AppBase/etc/iemp.framework/ip.white.list/ip-white-list-openapi.properties”下配置白名单。

表2-2 ip-white-list-openapi.properties 配置项说明

参数	如何理解……	必选/可选	如何设置……	生效方式
key	访问 OpenAPI 接口的第三方应用的 IP 地址白名单。	必选	**数据类型：**字符串 **取值范围：**IP@jetty.openapi，其中 IP 是点分网络地址格式。	保存生效

参数	如何理解……	必选/可选	如何设置……	生效方式
			缺省值：无	

配置举例：

```
10.1.1.1@jetty.openapi
10.1.1.2@jetty.openapi
```

设置用户权限

说明
角色“Openapi 用户组”拥有调用 Open API 接口的权限，可以操作所有网元，但不能操作前台页面。

由于 Open API 会进行用户鉴权，所以网管管理员需要为调用 Open API 的用户设置权限。方法如下：

步骤 1 在主菜单中选择“系统 > 系统管理 > 用户管理”。

步骤 2 在左侧导航区中选择“用户”。

步骤 3 在“用户”页面中，单击“创建”。

步骤 4 在新打开的页面中，设置基本信息。

1. 在“基本信息”步骤中，设置“用户名”。
2. 单击“设置”，设置“密码”和“确认密码”。单击“确定”。
3. 设置“帐号使用状态”为“启用”。
4. 设置“类型”为“OpenAPI 用户”。
5. 设置“描述”。
6. 单击“下一步”。

步骤 5 在新打开的页面中，设置所属角色为“Openapi 用户组”，使用户拥有使用 Open API 的权限。单击“下一步”。

步骤 6 在新打开的页面中，设置访问控制，使用户只能在特定时间段、从特定 IP（Internet Protocol）地址登录 eSight。单击“完成”。

----结束

注意
若调用安全管理的角色、用户接口，需要拥有“系统 > 用户管理”的操作权限；若调用安全管理的审计日志接口，需要拥有“系统 > 日志管理”的操作权限。

由于调用安全管理的角色、用户接口或查询审计日志接口需要拥有新的操作权限，而角色“Openapi 用户组”的操作权限无法编辑，所以网管管理员需要为调用 Open API

的用户添加新的角色并设置操作权限。以添加“系统 > 用户管理”的操作权限为例，方法如下：

步骤 1 在主菜单中选择“系统 > 系统管理 > 用户管理”。

步骤 2 在左侧导航区中选择“角色”。

步骤 3 在“角色”页面中，单击“创建”。

步骤 4 在新打开的页面中，设置基本信息。

1. 在“基本信息”步骤中，设置“角色名”、“描述”。
2. 在用户列表中选择上一操作中步骤 4 创建的用户。
3. 单击“下一步”。

步骤 5 单击“下一步”。

步骤 6 选择操作，使该角色拥有指定的操作权限。

1. 在“选择操作”步骤中，单击“增加”。
2. 在“选择操作”对话框中选择操作权限“系统 > 用户管理”。单击“确定”。
3. 单击“下一步”。

步骤 7 确认结果。

1. 在“结果确认”步骤中，核对角色创建的所有信息。
2. 单击“完成”。

----结束

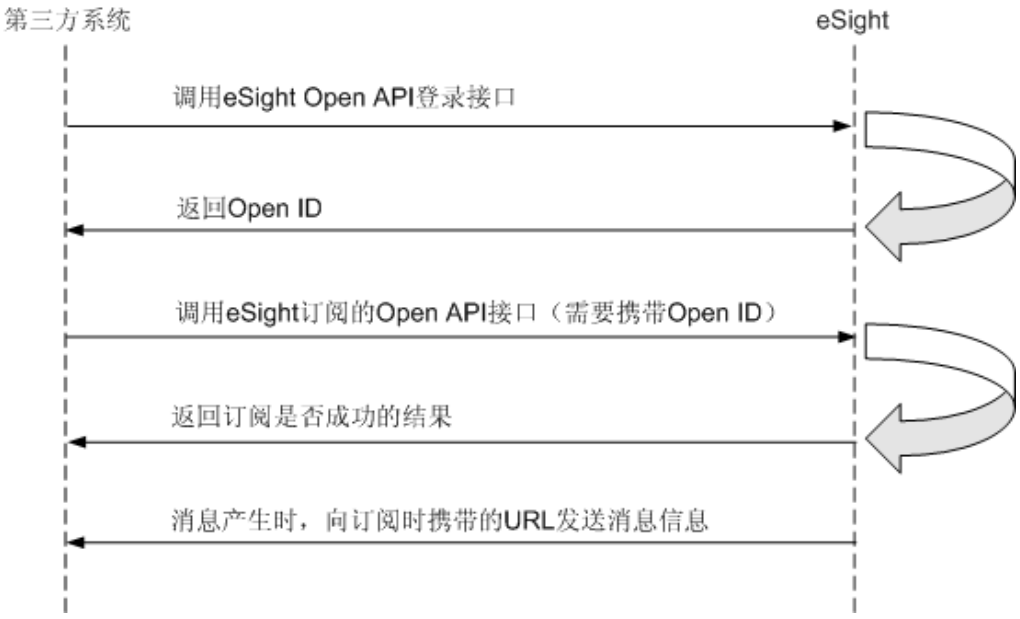
准备 WEB 消息通知的环境

第三方系统为了接收 eSight 系统以 WEB 方式发送的消息通知，必须作为 WEB 服务器，实现处理消息通知请求的 WEB Servlet，能够接收 HTTPS 或 HTTP 请求并回应执行结果。

首先以 OpenAPI 用户调用 OpenAPI 登录接口登录 eSight 系统获取 openID，然后订阅待接收处理的消息通知，在订阅消息中必须携带第三方系统的系统标识、接收处理消息通知的 WEB Servlet URL 和第三方系统用于鉴别接收到的请求报文有效性的 openID。

eSight V300R006C00 Open API 用户指南

图2-2 WEB 消息通知流程图



1. 订阅接口必须携带第三方系统的唯一标识符 systemID，eSight 系统根据唯一标识符 systemID 识别第三方系统，如果第三方系统调用两次 eSight 资源变更订阅接口，且 systemID 相同，则第二次调用时为更新订阅信息。
2. 当第三方系统的信息（OpenID，url）发生改变时，需要重新订阅，每次重新订阅仅更新订阅信息，待发送的消息队列中缓存的消息不丢弃。第三方系统接收消息通知的 URL 和 openID 的有效性由第三方系统保证。
3. 当第三方系统不希望再处理某类消息通知或者需要清除还未接收处理的 eSight 系统中的待发送消息，需要调用退订接口，eSight 系统将清除原订阅接口中的信息和待发送消息队列中的所有消息，后续 eSight 系统此类资源发送变更，将不再通知到第三方系统。
4. eSight 系统可以支持使用 HTTPS 或者 HTTP 协议向第三方系统发送变更通知。当第三方系统订阅接口中注册的接收消息通知的 URL 为 HTTPS 协议，eSight 系统与第三方系统使用匿名认证方式建立 SSL 连接，HTTP 协议报文体中 data 属性不加密，依靠 HTTP 报文传输通道加密确保数据的安全性。如果第三方系统接收消息的 URL 为 HTTP 协议，eSight 系统将对 HTTP 报文的 data 字段采用 AES128 加密算法对数据进行加密，第三方系统接收到数据后需要对数据解密后才能使用。AES128 加密密钥见下表对配置文件的说明。

   位置：<安装目录>/AppBase/etc/oms.ros/ros.web.notification.xml

表2-3 ros.web.notification.xml 配置项说明

参数	如何理解……	必选/可选	如何设置……	生效方式
enable-ip-white-list	是否启用 IP 白名单方式允许 eSight 系统向那些 IP 地址发送消息通知，与 OpenAPI 开放接口使用同一个白名单，	必选	**数据类型**：布尔型 **取值范围**：false 或 true	重启生效

eSight V300R006C00 Open API 用户指南

参数	如何理解……	必选/可选	如何设置……	生效方式
	配置文件位置为：etc/iemp.framework/ip.white.list/ip-white-list-openapi.properties。 说明 白名单配置文件修改后保存即生效，无需重启。		**缺省值：**true	
http-encryption.key	向第三方系统以明文HTTP 协议发送消息通知对报文体中 data 字段执行 AES128 加密的密钥，必须为 16 进制表示 128bit（16 个字节）的字符串。 配置文件中以密文存储密钥，如果修改默认的明文密码，请使用/AppBase/tools/bmetool/encrypt/encrypt.bat 或 ncrypt.sh 加密工具加密，同时更新配置文件，注意加密工具要求待加密明文字符串至少包含数字、大小字母。	必选	**数据类型：**String **取值范围：**32 个字符 **每字符取值：**0~9A~F **缺省值：**对应明文缺省密钥为E5DF96DE53ED56CBAE25368624DC25Ef	重启生效
http-encryption.iv	向第三方系统以明文HTTP 协议发送消息通知对报文体中 data 字段执行 AES128 加密的随机数发生器 IV 值，必须为 16 进制表示 128bit（16 个字节）的字符串。 配置文件中以密文存储密文，如果修改默认的明文 IV 值，请使用/AppBase/tools/bmetool/encrypt/encrypt.bat 或 ncrypt.sh 加密工具加密，同时更新配置文件，注意加密工具要求待加密明文字符串至少包含数字、大小字母。	必选	**数据类型：**String **取值范围：**32 个字符 **每字符取值：**0~9A~F **缺省值：**对应明文缺省值为E2CDBCA5693654EF2E6F38693653564E	重启生效
message-queue.queue.size	每一个第三方系统待发送消息队列最大长度。	必选	**数据类型：**整数 **取值范围：**	重启生效

参数	如何理解……	必选/可选	如何设置……	生效方式
			1~30000 **缺省值：**10000	
message-queue.queue.overflow.alarm-threshold-percent	每一个第三方系统待发送消息队列溢出预警阈值，单位：百分比。	必选	**数据类型：**整数 **取值范围：**1~100 **缺省值：95**	重启生效
message-queue.queue.overflow.strategy	每一个第三方系统待发送消息队列溢出后对消息处理策略。	必选	**数据类型：**枚举 **取值范围：**discard_new_event 和 store_in_temp_file **缺省值：**discard_new_event	重启生效
message-queue.sender.response-timeout-second	向第三方系统发送消息通知的超时时间，单位：秒。	必选	**数据类型：**整数 **取值范围：**1~60 **缺省值：**3	重启生效
message-queue.sender.retry-times	向第三方系统发送消息通知的重发次数。	必选	**数据类型：**整数 **取值范围：**1~10 **缺省值：**3	重启生效
message-queue.sender.send-interval-millisecond	向第三方系统发送消息通知的连续消息之间的间隔时间，单位：毫秒。	必选	**数据类型：**整数 **取值范围：**1~60000 **缺省值：**0	重启生效
message-queue.keep-alive.interval-minute	eSight 系统在指定时间内未与第三方系统交互主动发送保活消息，单位：分钟。	必选	**数据类型：**整数 **取值范围：**1~30 **缺省值：**5	重启生效
platform.version-file	产品版本文件的位置，以 appbase/etc 作为根路径的相对路径。	必选	**数据类型：**String **取值范围：**文	重启生效

eSight V300R006C00 Open API 用户指南

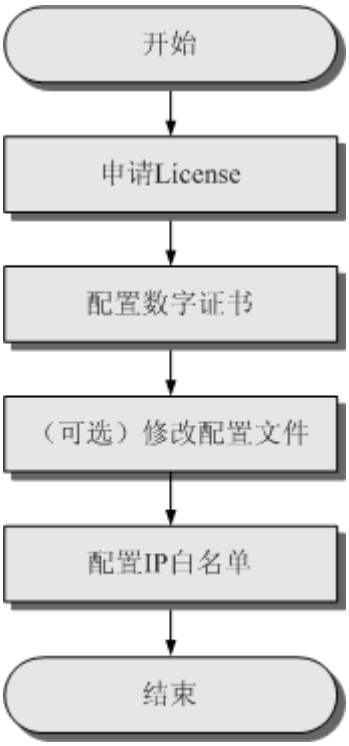
参数	如何理解……	必选/可选	如何设置……	生效方式
			件名称 **缺省值：**platformversion.xml	
platform.id	产品 ID。	必选	**数据类型：**String **取值范围：**可见字符串 **缺省值：**HuaweiPlatform	重启生效
platform.name	产品版本文件中读取产品版本信息的属性名称。	必选	**数据类型：**String **取值范围：**可见字符串 **缺省值：**PLATFORM_VERSION	重启生效
max-system-number	可接收 eSight 消息通知的第三方系统最大个数。	必选	**数据类型：**整数 **取值范围：**1~10 **缺省值：**10	重启生效
system-list.system.id	每一个第三方系统均配置，系统标识。	必选	**数据类型：**字符串 **取值范围：**0-9a-zA-Z@_-(),.^$~`！最大长度 64 **缺省值：**无	重启生效
system-list.system.system-up-url	每一个第三方系统均配置，接收 eSight 系统上电通知的 URL（不应该执行 openID 认证）。	必选	**数据类型：**String **取值范围：**URL **缺省值：**无	重启生效
system-list.system.max-event-aggregation-number	每一个第三方系统均配置，同类资源同消息类型的消息通知汇聚在一个 HTTP 报文中发送，最大汇聚个数。	必选	**数据类型：**整数 **取值范围：**1~30 **缺省值：**5	重启生效

eSight V300R006C00 Open API 用户指南

2.2 基于摘要认证

请参考图 2-3 所示做好基于摘要认证的前期准备。

图2-3 前期准备



申请 License

网管用户需要提前向华为公司申购 License，否则代理商无法调用 Open API。

用户申请的 License 控制项为“SNMP 告警北向功能&OpenSDK”（编号为 LNSHFSDK031）。

申请 License 的详细流程请参考《License 申请指导书》。

查看当前 License 是否有“SNMP 告警北向功能&OpenSDK”控制项的方法如下：

1. 在主菜单中选择“系统 > 系统管理 > License 管理”。
2. 在“License 管理”页面中，可以查看 License 基本信息、资源控制信息、功能控制信息和失效码信息。

配置数字证书

调用 Open API 是采用 HTTPS 方式，所以在建立 HTTPS 请求连接时需要证书。证书的默认路径是：<安装目录>/AppBase/etc/certificate/JettyServerKeyStore。

注意

- eSight 不提供证书，需要企业用户自己生成证书放至默认路径下。
- 请确保生成的证书密码与 ros.xml 中的配置项 keystorePass 一致；生成的证书存放路径与 ros.xml 中的配置项 keystorefilePath 一致。配置方法可参见表 2-4。

修改配置文件

1. 网管管理员需要参照下表设置“<安装目录>/AppBase/etc/oms.ros/ros.xml”文件中的参数。

表2-4 ros.xml 配置项说明

参数	如何理解……	必选/可选	如何设置……	生效方式
webservers/rosOpenAPIROA/threadpool.thread.max	线程池最大线程数。	可选	**数据类型：** String **参数值：** 数字字符串 **缺省值：** 100	重启生效
webservers/rosOpenAPIROA/threadpool.queue.max	线程池队列最大值。	可选	**数据类型：** String **参数值：** 数字字符串 **缺省值：** 2000	重启生效
webservers/rosOpenAPIROA/roa.server.ip.white.list.enable	是否开启 IP 白名单。	可选	**数据类型：** 布尔型 **取值范围：** false 或 true **缺省值：** true	重启生效
webservers/rosOpenAPIROA/ip.white.type	白名单标识。	可选	**数据类型：** String **取值范围：** 字符串 **缺省值：** jetty.openapi	重启生效
webservers/rosOpenAPIROA/	Open API 接口调用时的	必选	**数据类型：**	重启生效

参数	如何理解……	必选/可选	如何设置……	生效方式
connectors/openapiROAConnector/ip	服务器 IP 地址。 说明 在实际使用中，需要将该 IP 地址替换为网管的 IP 地址（不能为 127.0.0.1 或 0.0.0.0），第三方才能调用 Open API 接口。否则，只有本机才能调用 Open API 接口。		String **参数值：**数字字符串表示的 IP 地址 **缺省值:**127.0.0.1	
webservers/rosOpenAPIROA/connectors/openapiROAConnector/port	调用 Open API 时的端口号。	必选	**数据类型：**String **参数值：**数字字符串表示的端口号 **缺省值：**32102	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.protocol	OpenAPI 接口调用时的协议版本。 说明 • 该配置项默认不提供（此时默认支持 TLSv1.1 和 TLSv1.2），由用户自行配置。 • 由于 jre1.7 不完全支持 TLSv1.2，若使用 TLSv1.2 协议，推荐使用 jre1.8。	可选	**数据类型：**String **参数值：**支持使用的 https 协议版本 **缺省值：**无	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.keystore.path	HTTPS 连接时证书的存放路径。	必选	**数据类型：**String **取值范围：**路径字符串 **缺省值：**etc/certificate/JettyServerKeyStore	重启生效
webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.keystore.password	HTTPS 连接时证书的密码。 说明 该密码是使用 encrypt 生成的密文密码。 • 在 Windows 操作系统中，需要使用“<安装目录>/AppBase/tools/bmetool/encrypt/encrypt.bat”工具来加密存储密	必选	**数据类型：**String **参数值：**字符串表示的证书密码 **缺省值：**9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555	重启生效

eSight V300R006C00 Open API 用户指南

参数	如何理解……	必选/可选	如何设置……	生效方式
	码，命令格式为encrypt.bat 0。 • 在Linux操作系统中，需要使用“<安装目录>/AppBase/tools/bmetool/encrypt/encrypt.sh”工具来加密存储密码，命令格式为./encrypt.sh 0。		fd5aff9908d	

配置举例：

```
<?xml version="1.0" encoding="UTF-8"?>
<webservers>
  <webserver name="rosOpenAPIROA">
    <property name="threadpool.thread.max" value="100" />
    <property name="threadpool.queue.max" value="2000" />
    <property name="roa.server.ip.white.list.enable" value="true" />
    <property name="ip.white.type" value="jetty.openapi" />
    <connectors>
       <connector name="openapiROAConnector" type="https">
          <property name="ip" value="127.0.0.1" />
          <property name="port" value="32102" />
          <property name="ssl.keystore.path"
value="etc/certificate/JettyServerKeyStore" />
          <property name="ssl.keystore.password"
value="9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555fd5aff9908d" />
       </connector>
     </connectors>
   </webserver>
</webservers>
```

2. 网管管理员需要参照下表设置“<安装目录>/AppBase/etc/oms.ros/rosinternal.xml”文件中的参数。

表2-5 rosinternal.xml 配置项说明

参数	如何理解……	必选/可选	如何设置……	生效方式
as/ip	分开部署时服务端的IP地址。	可选	**数据类型：** String **取值范围：** 数字字符串 **缺省值：** 无	重启生效
as/port	分开部署时服务端的端口。	可选	**数据类型：** String **取值范围：** 数字字符串	重启生效

eSight V300R006C00 Open API 用户指南

参数	如何理解……	必选/可选	如何设置……	生效方式
			缺省值：无	
as/openapiServicePort	OpenAPI 服务所在 ROA 容器的端口。	可选	**数据类型**：String **取值范围**：数字字符串 **缺省值**：32102	重启生效
as/tokenTimeout	token 失效的时间。	必选	**数据类型**：String **取值范围**：数字字符串，1800~14400 **缺省值**：14400 **单位**：秒	重启生效
securityManager/lockIntervalTime	这 3 个配置项需要配合使用配置调用 OpenAPI 接口的锁定策略，单位分别为秒、次、秒。 例如 3 个配置项的配置值如下： lockIntervalTime：600 errMaxCount：5 errIntervalTime：300 则表示用户在 300 秒之内调用 OpenAPI 错误次数达到 5 次，那么 eSight 会锁定 600 秒不支持调用 openapi 。	必选	**数据类型**：String **取值范围**：数字字符串，30~14400 **缺省值**：600 **单位**：秒	重启生效
securityManager/errMaxCount		必选	**数据类型**：String **取值范围**：数字字符串，1~30 **缺省值**：5 **单位**：次	重启生效
securityManager/errIntervalTime		必选	**数据类型**：String **取值范围**：数字字符串，30~14400 **缺省值**：300 **单位**：秒	重启生效
callCondionSet/methodResponseTime	接口调用的响应时间。	可选	**数据类型**：String **取值范围**：数字字符串，100~10000	重启生效

eSight V300R006C00 Open API 用户指南

参数	如何理解……	必选/可选	如何设置……	生效方式
			缺省值：2000	
callCondionSet/perMethodPerMinuteCallTime	接口每分钟允许调用的次数。	可选	**数据类型**：String **取值范围**：数字字符串，400~1000 **缺省值**：400	重启生效
callCondionSet/requestLogmaxSize	接口调用统计的最大日志条数。	可选	**数据类型**：String **取值范围**：数字字符串，1~20000 **缺省值**：20000	重启生效
callCondionSet/logDeleteRecord	达到接口调用统计的最大日志条数后每次删除的日志条数。	可选	**数据类型**：String **取值范围**：数字字符串，500~5000 **缺省值**：500	重启生效
digistAuthUsers/xx	摘要认证的用户名，该配置项的节点值为该用户名对应的密码。 说明 • 该配置项默认不提供，只在摘要认证时需要用户自行配置。 • 用户名及倒序不能和密码相同。 • 用户名不能小于 6 个字符,不能大于 32 个字符,不能包含空格和"\#%/&';<+=>?©® 特殊字符。 • 密码需要满足密码复杂度。 该用户密码是使用 encrypt 生成的密文密码。 • 在 Windows 操作系统中，需要使用“<安装目录>/AppBase/tools/bmetool/encrypt/encrypt.bat”工具来加密存储密码，命令格式为	可选	**数据类型**：String **取值范围**：不受限制 **缺省值**：无	重启生效

eSight V300R006C00 Open API 用户指南

参数	如何理解……	必选/可选	如何设置……	生效方式
	encrypt.bat 0。 • 在 Linux 操作系统中，需要使用“<安装目录>/AppBase/tools/bmetool/encrypt/encrypt.sh”工具来加密存储密码，命令格式为./encrypt.sh 0。			

配置举例：

```
<?xml version="1.0" encoding="UTF-8"?>

<config name="oms">
 <config name="as">
  <!--ip of default roa service-->
  <param name="ip"></param>
  <!--https port of default roa service-->
  <param name="port"></param>
  <!--https port of internal openapi service-->
  <param name="openapiServicePort">32102</param>
  <!--token timeout (second), min: 1800, max: 14400-->
  <param name="tokenTimeout">14400</param>
 </config>
 <config name="securityManager">
    <!--account lock time (second), min: 30, max: 14400-->
  <param name="lockIntervalTime">600</param>
    <!--max retry times, min: 1, max: 30-->
  <param name="errMaxCount">5</param>
    <!--retry interval time (second), min: 30, max: 14400-->
  <param name="errIntervalTime">300</param>
 </config>
 <config name="callCondionSet">
  <param name="methodResponseTime">2000</param>
  <param name="perMethodPerMinuteCallTime">400</param>
  <param name="requestLogmaxSize">20000</param>
  <param name="logDeleteRecord">500</param>
 </config>
 <config name="digistAuthUsers">
  <param name="user1234">
9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555fd5aff9908d</param>
  <param name="user5678">
9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555fd5aff9908d</param>
 </config>
</config>
```

配置 IP 白名单

网管管理员需要参照下表配置“<安装目录>/AppBase/etc/iemp.framework/ip.white.list/ip-white-list-openapi.properties”文件中的白名单。

eSight V300R006C00 Open API 用户指南

说明
双机环境时，请网管管理员登录到主机路径“<安装目录>/eSightSync/AppBase/etc/iemp.framework/ip.white.list/ip-white-list-openapi.properties”下配置白名单。

表2-6 ip-white-list-openapi.properties 配置项说明

参数	如何理解……	必选/可选	如何设置……	生效方式
key	访问 OpenAPI 接口的第三方应用的 IP 地址白名单。	必选	**数据类型**：字符串 **取值范围**：IP@jetty.openapi，其中 IP 是点分网络地址格式。 **缺省值**：无	保存生效

配置举例：

```
10.1.1.1@jetty.openapi
10.1.1.2@jetty.openapi
```

eSight V300R006C00 Open API 用户指南

3 DEMO

本节通过一个 DEMO 介绍如何调用 Open API 接口开发应用。

3.1 开发一个基于 HttpClient 的应用

功能介绍

本应用展示如何调用基于 HttpClient 的接口查询告警详细信息。

文件介绍

本应用包含如下文件。可以单击 ros_demo.rar 下载 DEMO。

```
|-lib
|  └-com.springsource.org.apache.commons.lang-2.4.0.jar
|  └-commons-beanutils-1.8.0.jar
|  └-commons-collections-3.2.1.jar
|  └-commons-logging-1.1.1.jar
|  └-ezmorph-1.0.6.jar
|  └-httpclient-4.2.2.jar
|  └-httpcore-4.2.2.jar
|  └-json-lib-2.4-jdk15.jar
|-src
|  └-ros
|    └-basedhttpclient
|       └-ParseResponse.java
|       └-TestGetAlarms.java
```

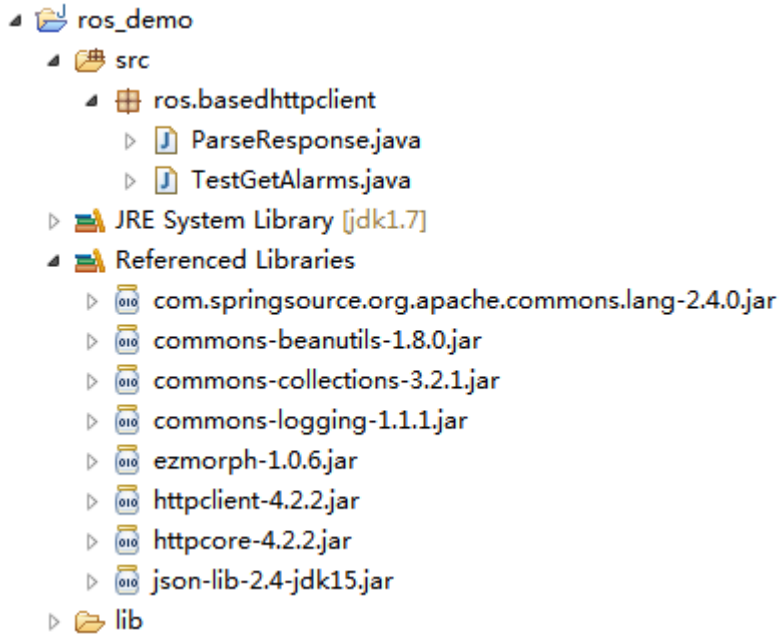
说明

- *.java 文件是本应用的源代码，编码格式为 UTF-8。main 函数在 TestGetAlarmDetail.java 文件中，ParseResponse.java 文件用于解析 JSON 字符串。
- *.jar 文件是本应用依赖的 jar 包。为了确保编译正确，请按照上述列表获取对应版本的 jar 包。

- 建议使用 JRE 1.7 或以上版本。

可以参考图 3-1 在 Eclipse 开发环境中创建本应用的工程。

图3-1 工程结构图



基于 openid 认证开发应用

说明

- 以下示例代码未对服务端证书做校验，仅供参考。在第三方开发过程中，最好校验证书，否则存在安全风险。
- 请确保客户端使用的协议版本与网管一致，否则接口调用不通。网管的协议版本配置请参考修改配置文件。

步骤 1 登录并获取 openid。

```
//set the URL
String openidURL = "/rest/openapi/sm/session";
//set parameters
List<BasicNameValuePair> parameters = new ArrayList<BasicNameValuePair>();
parameters.add(new BasicNameValuePair("userid", userName));
parameters.add(new BasicNameValuePair("value", pwd));
parameters.add(new BasicNameValuePair("ipaddr", "10.10.10.10"));

//create a connection manager
X509TrustManager tm = new X509TrustManager() {
 public void checkClientTrusted(X509Certificate[] xcs, String string) {
 }

 public void checkServerTrusted(X509Certificate[] xcs, String string) {
 }

 public X509Certificate[] getAcceptedIssuers() {
```

eSight V300R006C00 Open API 用户指南


```
   return null;
  }
 };

 //create a SSL connection
 SSLContext sslcontext = SSLContext.getInstance("TLS");
 sslcontext.init(null, new TrustManager[] { tm }, null);
 SSLSocketFactory socketFactory = new SSLSocketFactory(sslcontext,
SSLSocketFactory.ALLOW ALL HOSTNAME VERIFIER);

 SchemeRegistry schemeRegistry = new SchemeRegistry();
 schemeRegistry.register(new Scheme("https", port, socketFactory));

 //create a HttpClient to connect to the target host
 HttpClient httpClient = new DefaultHttpClient(new
BasicClientConnectionManager(schemeRegistry));

 //set the URL
 String url = "https://" + ip + ":" + port + openidURL;

 //set the method
 HttpPut httpPut = new HttpPut(url);
 httpPut.setEntity(new UrlEncodedFormEntity(parameters, "UTF-8"));

 //send the request
 HttpResponse response = httpClient.execute(httpPut);
 Map<String, String> retMap = ParseResponse.parseResponse(getResult(response));
 if (retMap.get("code").equals("0")) {
  return retMap.get("data");
 }
 return "";
```

步骤 2 调用接口。

```
 //set the URL
 String queryNeURL = "/rest/openapi/alarm";
 //set headers and parameters
 List<BasicNameValuePair> headers = new ArrayList<BasicNameValuePair>();
 headers.add(new BasicNameValuePair("openid", openid));
 List<BasicNameValuePair> parameters = new ArrayList<BasicNameValuePair>();
 parameters.add(new BasicNameValuePair("size", "100"));

 //create a connection manager
 X509TrustManager tm = new X509TrustManager() {
  public void checkClientTrusted(X509Certificate[] xcs, String string) {
  }

  public void checkServerTrusted(X509Certificate[] xcs, String string) {
  }

  public X509Certificate[] getAcceptedIssuers() {
   return null;
  }
 };

 //create a SSL connection
```

eSight V300R006C00 Open API 用户指南


```
 SSLContext sslcontext = SSLContext.getInstance("TLS");
 sslcontext.init(null, new TrustManager[] { tm }, null);
 SSLSocketFactory socketFactory = new SSLSocketFactory(sslcontext,
SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);

 SchemeRegistry schemeRegistry = new SchemeRegistry();
 schemeRegistry.register(new Scheme("https", port, socketFactory));

 //create a HttpClient to connect to the target host
 HttpClient httpClient = new DefaultHttpClient(new
BasicClientConnectionManager(schemeRegistry));

 //set the URL
 String url = "https://" + ip + ":" + port + queryNeURL;

 //set parameters
 if (null != parameters) {
  url += " ";
  boolean init = false;
  for (BasicNameValuePair e : parameters) {
   if (!init) {
    url += URLEncoder.encode(e.getName(), "UTF-8") + "=" +
URLEncoder.encode(e.getValue(), "UTF-8");
    init = true;
   } else {
    url += "&" + URLEncoder.encode(e.getName(), "UTF-8") + "=" +
URLEncoder.encode(e.getValue(), "UTF-8");
   }
  }
 }
    HttpGet httpGet = new HttpGet(url);
    //set headers
 if (null != headers) {
  for (BasicNameValuePair header : headers) {
   httpGet.setHeader(header.getName(), header.getValue());
  }
 }

 //send the request
 HttpResponse response = httpClient.execute(httpGet);
 String ret = getResult(response);
 if (null == ret || ret.isEmpty()) {
  return "";
 }

 //get the result
 Map<String, String> retMap = ParseResponse.parseResponse(ret);
 if (retMap.get("code").equals("0")) {
  return retMap.get("data");
 }
 return "";
```

----结束

eSight V300R006C00 Open API 用户指南

基于摘要认证开发应用

说明

- 开发的应用不能从上下文 HttpContext 中去获取参数。
- 必须调用 eSight 的注册 ROA 接口来注册 OpenAPI 服务。
- 以下示例代码未对服务端证书做校验，仅供参考。在第三方开发过程中，最好校验证书，否则存在安全风险。
- 请确保客户端使用的协议版本与网管一致，否则接口调用不通。网管的协议版本配置请参考修改配置文件。

步骤 1 获取随机数。

```
import javax.net.ssl.SSLContext;
import javax.net.ssl.X509TrustManager;
import javax.net.ssl.TrustManager;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.security.cert.X509Certificate;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpDelete;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.methods.HttpPut;
import org.apache.http.client.methods.HttpUriRequest;
import org.apache.http.conn.scheme.Scheme;
import org.apache.http.conn.scheme.SchemeRegistry;
import org.apache.http.conn.ssl.SSLSocketFactory;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.impl.conn.BasicClientConnectionManager;
import org.apache.http.message.BasicNameValuePair;

public class NewHttpsAccess {
   /**
    * http 调用
    * @param ip           IP
    * @param port           端口
    * @param url         URL 地址
    * @throws Exception
    */
   public static HttpResponse access(String ip, int port, String url, String method,
BasicNameValuePair[] headers, BasicNameValuePair[] parameters) throws Exception {
      List<BasicNameValuePair> headers2 = null;
      List<BasicNameValuePair> parameters2 = null;
      if (null != headers) {
         headers2 = new ArrayList<BasicNameValuePair>();
```

eSight V300R006C00 Open API 用户指南


```
            for (BasicNameValuePair e : headers) {
                headers2.add(e);
            }
        }
        if (null != parameters) {
            parameters2 = new ArrayList<BasicNameValuePair>();
            for (BasicNameValuePair e : parameters) {
                parameters2.add(e);
            }
        }
        return access(ip, port, url, method, headers2, parameters2);
    }

    /**
     * http 调用
     * @param ip           IP
     * @param port           端口
     * @param url         URL 地址
     * @throws Exception
     */
    public static HttpResponse access(String ip, int port, String url, String method,
List<BasicNameValuePair> headers, List<BasicNameValuePair> parameters) throws
Exception {
        X509TrustManager tm = new X509TrustManager() {
            public void checkClientTrusted(X509Certificate[] xcs, String string) {
            }

            public void checkServerTrusted(X509Certificate[] xcs, String string) {
            }

            public X509Certificate[] getAcceptedIssuers() {
                return null;
            }
        };

        SSLContext sslcontext = SSLContext.getInstance("TLS");
        sslcontext.init(null, new TrustManager[] { tm }, null);
        SSLSocketFactory socketFactory = new SSLSocketFactory(sslcontext,
SSLSocketFactory.ALLOW_ALL_HOSTNAME_VERIFIER);

        SchemeRegistry schemeRegistry = new SchemeRegistry();
        schemeRegistry.register(new Scheme("https", port, socketFactory));

        HttpClient httpClient = new DefaultHttpClient(new
BasicClientConnectionManager(schemeRegistry));

        // URL
        url = "https://" + ip + ":" + port + url;
        System.out.println(url);

        //设置参数
        HttpUriRequest httpRequest = null;
        if ("PUT".equalsIgnoreCase(method)) {
            HttpPut httpPut = new HttpPut(url);
            httpRequest = httpPut;
```

eSight V300R006C00 Open API 用户指南


```
          if (null != parameters) {
             try {
                UrlEncodedFormEntity tmp = new UrlEncodedFormEntity(parameters,
"UTF-8");
                httpPut.setEntity(tmp);
             } catch (UnsupportedEncodingException e1) {
                e1.printStackTrace();
             }
          }
       } else if ("POST".equalsIgnoreCase(method)) {
          HttpPost httpPost = new HttpPost(url);
          httpRequest = httpPost;
          if (null != parameters) {
             try {
                UrlEncodedFormEntity tmp = new UrlEncodedFormEntity(parameters,
"UTF-8");
                httpPost.setEntity(tmp);
             } catch (UnsupportedEncodingException e1) {
                e1.printStackTrace();
             }
          }
       } else if ("GET".equalsIgnoreCase(method)) {
          if (null != parameters) {
             url += "?";
             boolean init = false;
             for (BasicNameValuePair e : parameters) {
                if (!init) {
                   url += URLEncoder.encode(e.getName(), "UTF-8") + "=" +
URLEncoder.encode(e.getValue(), "UTF-8");
                   init = true;
                } else {
                   url += "&" + URLEncoder.encode(e.getName(), "UTF-8") + "=" +
URLEncoder.encode(e.getValue(), "UTF-8");
                }
             }
          }
          HttpGet httpGet = new HttpGet(url);
          httpRequest = httpGet;
       } else if ("DELETE".equalsIgnoreCase(method)) {
          if (null != parameters) {
             url += "?";
             boolean init = false;
             for (BasicNameValuePair e : parameters) {
                if (!init) {
                   url += URLEncoder.encode(e.getName(), "UTF-8") + "=" +
URLEncoder.encode(e.getValue(), "UTF-8");
                   init = true;
                } else {
                   url += "&" + URLEncoder.encode(e.getName(), "UTF-8") + "=" +
URLEncoder.encode(e.getValue(), "UTF-8");
                }
             }
          }
          HttpDelete httpDelete = new HttpDelete(url);
          httpRequest = httpDelete;
```

eSight V300R006C00 Open API 用户指南


```
        } else {
            return null;
        }

        //设置 header
        if (null != headers) {
            for (BasicNameValuePair header : headers) {
                httpRequest.setHeader(header.getName(), header.getValue());
            }
        }

        // 发送请求
        HttpResponse response = httpClient.execute(httpRequest);
        return response;
    }

    public static String getResult(HttpResponse response) throws
IllegalStateException, IOException {
        // 输出返回值
        InputStream is = response.getEntity().getContent();
        BufferedReader br = new BufferedReader(new InputStreamReader(is));
        String ret = "";
        String line = "";
        while ((line = br.readLine()) != null) {
            if (!ret.isEmpty()) {
                ret += "\n";
            }
            ret += line;
        }
        return ret;
    }
}
    private static final String GLOBAL IP = "10.66.90.144";
    private static final int GLOBAL PORT = 32102;

    @Test
    public void correct nedevice queryby Test() throws Exception
    {
        String openidURL = "/rest/openapi/netype";
        String method = "GET";
        //摘要用户名
        String DIGEST USER = "user1234";
        long nounce = getNounceFromRequest(openidURL, method, DIGEST USER);

        //摘要密码
        String DIGEST PASSWORD = "Test1234";
        //随机数记数
        String DIGEST NC = "1";
        //客户端随机数
        String DIGEST CLIENTNOUNCE = "0A4F113B";

        icorrect nedevice queryby Test(openidURL, method, nounce, DIGEST USER,
DIGEST PASSWORD, DIGEST NC, DIGEST CLIENTNOUNCE);
    }
```

eSight V300R006C00 Open API 用户指南


```
   private static long getNounceFromRequest(String url, String method, String
userName)  throws Exception{
       BasicNameValuePair[] headers =
       {
          new BasicNameValuePair("nounce", "0"),  //随机数填写 0，那么是申请新的随机数
          new BasicNameValuePair("digest", ""),
          new BasicNameValuePair("appkey", userName),
       };

       HttpResponse response = NewHttpsAccess.access(GLOBAL IP, GLOBAL PORT, url,
method, headers, null);
       String body = NewHttpsAccess.getResult(response);
       System.out.println(body);
       return saveNounce(body);
   }

   public static long saveNounce(String body) {
       if (null != body && !body.isEmpty()) {
          final String splitstr = "\"data\":\"";
          int nIndex = body.indexOf(splitstr);
          String data = null;
          if (nIndex >= 0) {
             String strtmp = body.substring(nIndex + splitstr.length());
             int nIndex2 = strtmp.indexOf('\"');
             if (nIndex2 > 0) {
                data = strtmp.substring(0, nIndex2);
             }
          }

          if (null == data || data.isEmpty()) {
             return 0;
          }

          return Long.valueOf(data);
       }

       return 0;
   }
```

步骤 2 计算摘要。

```
private static BasicNameValuePair[] getRosDigestHeader(String url, String method,
long nounce, String DIGEST_USER, String DIGEST_PASSWORD, String DIGEST_NC, String
DIGEST_CLIENTNOUNCE) throws Exception {

       String disgest = getDigestSrcData(DIGEST_USER,
              url,
              method,
              (nounce + "").toUpperCase(),
              DIGEST_NC,
              DIGEST_CLIENTNOUNCE,
              DIGEST_PASSWORD
              );
       System.out.println(disgest);
       BasicNameValuePair[] headers =
       {
```

eSight V300R006C00 Open API 用户指南


```
                new BasicNameValuePair("nounce", nounce + ""),
                new BasicNameValuePair("digest", disgest),
                new BasicNameValuePair("appkey", DIGEST USER),
                new BasicNameValuePair("digestAlgorithm", "SHA256"),
                new BasicNameValuePair("nc", DIGEST NC),
                new BasicNameValuePair("clientNounce", DIGEST CLIENTNOUNCE),
        };

        return headers;
    }

    /**
     * 返回计算摘要的原始数据
     * 如果随机数失效，那么返回 null
     * If the algorithm directive's value is "MD5-sess", then HA1 is
     *      HA1=MD5(MD5(username:realm:password):nonce:cnonce)
     * If the qop directive's value is "auth" or is unspecified, then HA2 is
     *      HA2=MD5(method:digestURI)
     * If the qop directive's value is "auth" or "auth-int", then compute the
response as follows:
     *      response=MD5(HA1:nonce:nonceCount:clientNonce:qop:HA2)
     *
     * @return 计算摘要的原始数据
     * @throws Exception
     */
    public static String getDigestSrcData(String appkey, String url, String method,
String nounce, String nc, String clientNounce, String appsecure) throws Exception {
        String qop = "auth";
        String realm = "openapi";

        //计算 HA1
        String str1 = appkey + ":" + realm + ":" + appsecure;
        String tmp = hashCalc(str1, appsecure);
        String str2 = tmp + ":" + nc + ":" + clientNounce;
        String ha1 = hashCalc(str2, appsecure);

        //计算 HA2
        String str3 = method + ":" + url;
        String ha2 = hashCalc(str3, appsecure);

        String rtn = hashCalc(ha1 + ":" + nounce + ":" + nc + ":" + clientNounce +
":" + qop + ":" + ha2, appsecure);
        return rtn;
    }

    private static String hashCalc(String str, String appsecure) throws Exception {
        String rtn = EncryptionUtil.hmacSHA256Encrypt(appsecure, str);
        return rtn;
    }
```

步骤 3 调用接口。

```
private void icorrect_nedevice_queryby_Test(String url, String method, long nounce,
String DIGEST USER, String DIGEST PASSWORD, String DIGEST NC, String
DIGEST CLIENTNOUNCE) throws Exception
    {
```

eSight V300R006C00 Open API 用户指南


```
        BasicNameValuePair[] headers = getRosDigestHeader(url, method, nounce,
DIGEST_USER, DIGEST_PASSWORD, DIGEST_NC, DIGEST_CLIENTNOUNCE);

        /*
         *  necategory    设备分类
            netype        设备类型
            neip        设备 ip 地址(ip 通过分隔)
            nestate        网络设备状态(连接状态)
            start        指定从哪个起始记录位置开始返回查询结果集。
            size        指定返回查询结果集总数。
            orderby        指定查询结果集采用的排序字段。
            desc        指定查询结果是否按降序排序。
            */
        BasicNameValuePair[] parameters =
        {
                new BasicNameValuePair("necategory", ""),
                new BasicNameValuePair("netype", ""),
                new BasicNameValuePair("neip", ""),
                new BasicNameValuePair("nestate", ""),
                new BasicNameValuePair("start", ""),
                new BasicNameValuePair("size", ""),
                new BasicNameValuePair("orderby", ""),
                new BasicNameValuePair("desc", "")
        };
        HttpResponse response = NewHttpsAccess.access(GLOBAL_IP, GLOBAL_PORT, url,
method,
                headers, parameters);
        String body = NewHttpsAccess.getResult(response);
        System.out.println(body);
    }
```

----结束

3.2 使用 Restclient 工具验证 Open API（仅限 openid 认证）

可以使用第三方工具 Restclient 来验证 Open API 接口。下载地址：
http://code.google.com/p/rest-client/downloads/detail?name=restclient-ui-3.2.2-jar-with-dependencies.jar&can=2&q=rest-client

说明

- 请确保客户端使用的协议版本与网管一致，否则接口调用不通。网管的协议版本配置请参考修改配置文件。
- 由于 Restclient 工具自身的限制，在使用 Restclient 工具验证 Open API 前，请在服务端的“<安装目录>/AppBase/etc/oms.ros/ros.xml”文件中添加 ssl.protocol 项（配置项路径为“webservers/rosOpenAPIROA/connectors/openapiROAConnector/ssl.protocol”），并重启服务端。

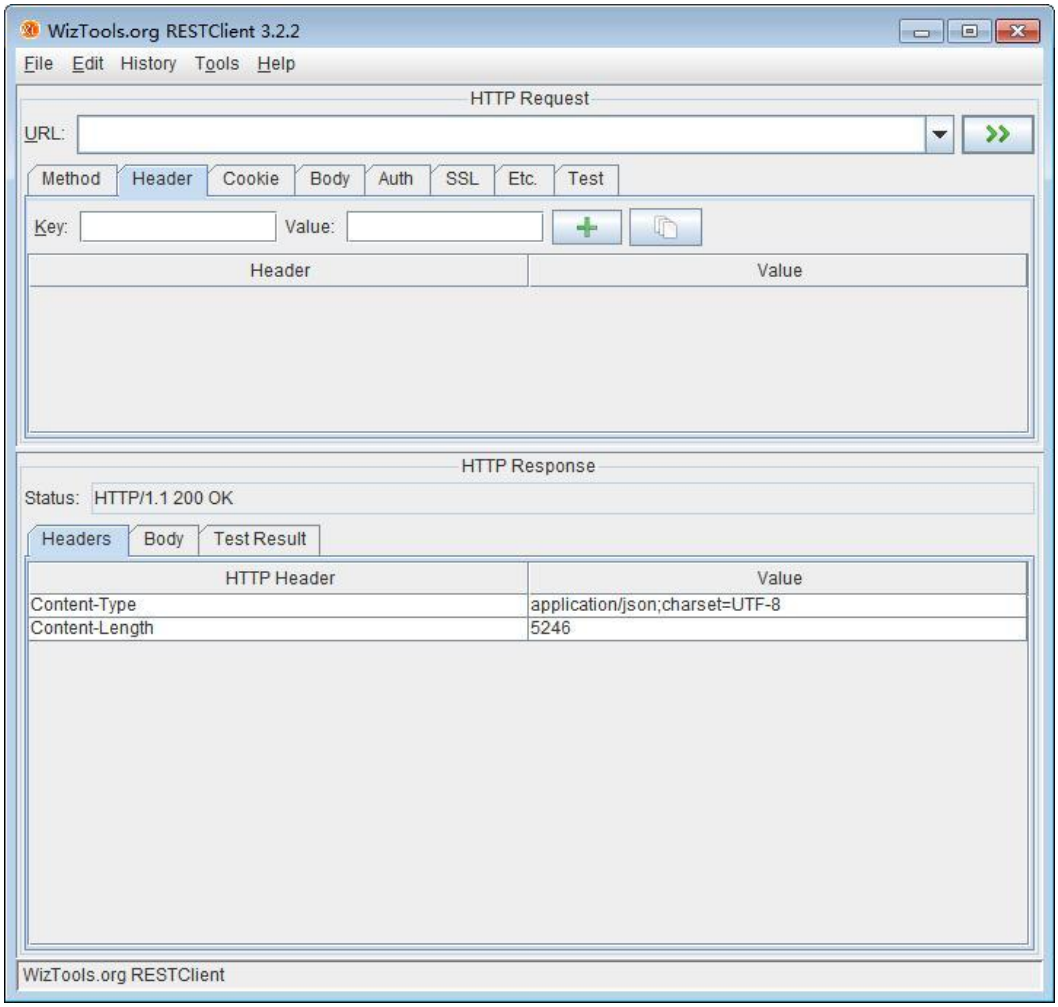
  ```
  <property name="ssl.protocol" value="SSLv2Hello;TLSv1.1;TLSv1.2;TLSv1" />
  ```

运行工具

在 JAR 包上单击鼠标右键，使用 Java 打开此程序。

eSight V300R006C00 Open API 用户指南

说明
需要运行环境已经安装了 JRE。



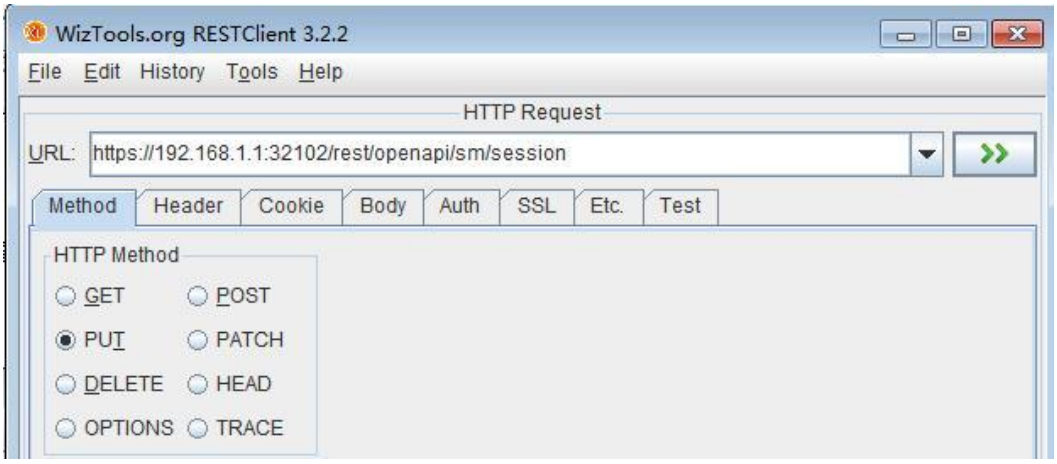
获取 openid

步骤 1 输入 URL。

格式为 https://IP:32102/rest/openapi/sm/session，其中 IP 是 Open API 服务所在机器的 IP 地址。如，https://192.168.1.1:32102/rest/openapi/sm/session。

步骤 2 在 Method 页签中选择“PUT”。

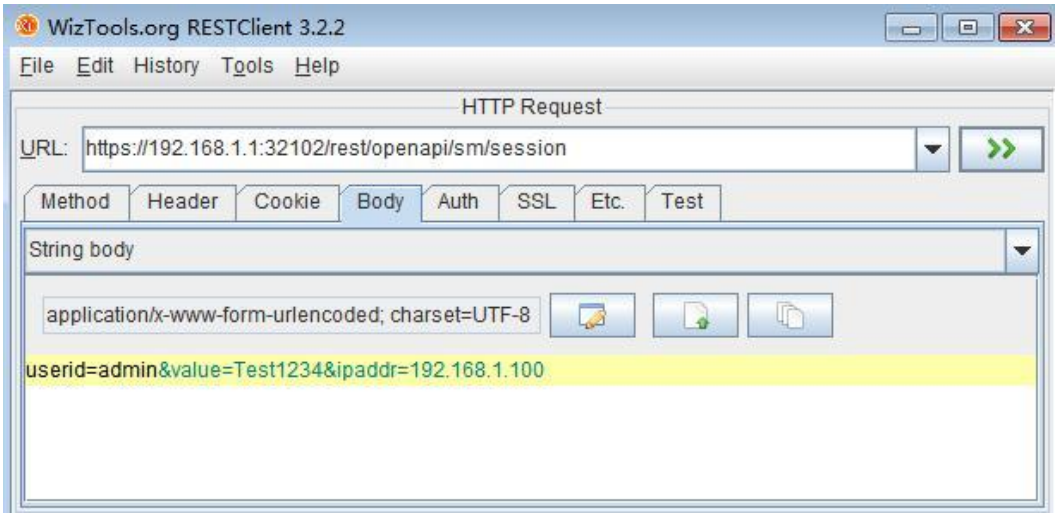
eSight V300R006C00 Open API 用户指南



步骤 3　在 Body 页签中输入参数，参数填写方法可以参考**错误！未找到引用源。错误！未找到引用源。**。

Body 类型为“String body”，报文类型为“application/x-www-form-urlencoded; charset=UTF-8”。

body 内容为“userid=admin&value=Test1234&ipaddr=192.168.1.100”，其中 userid、value、ipaddr 是参数名，在接口中有介绍。admin、Test1234、192.168.1.100 是值。

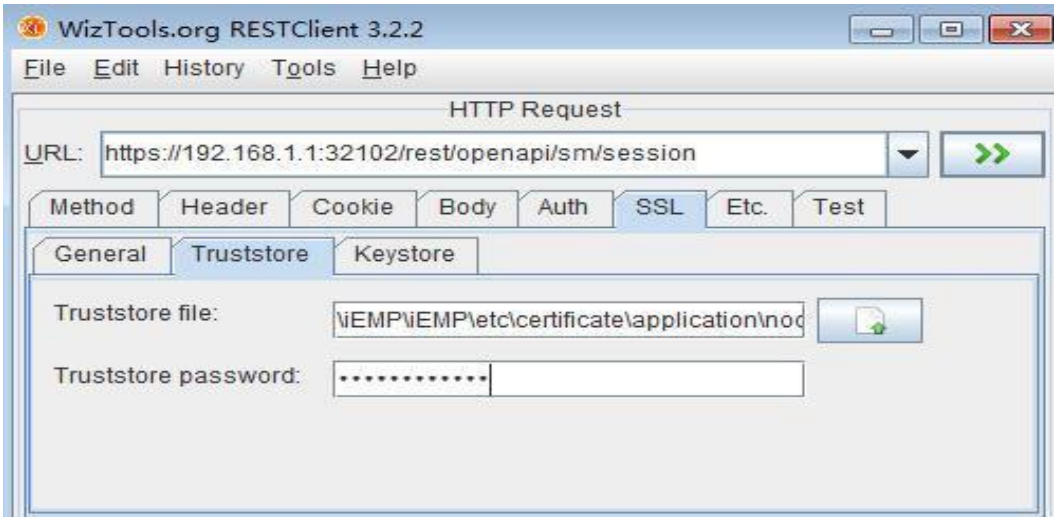


步骤 4　在 SSL 页签的 Truststore 标签中输入证书路径和密码。

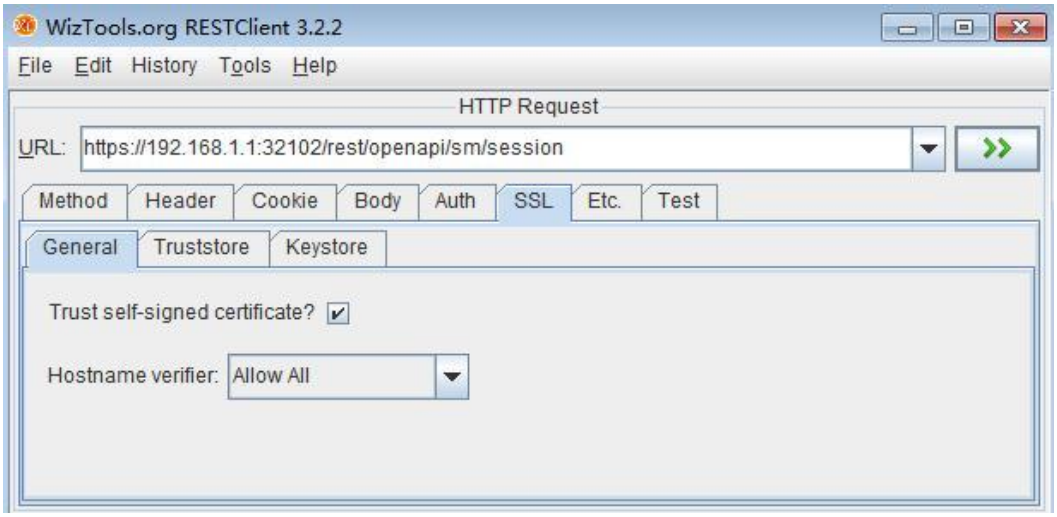
证书可以从网管获取，路径可以参考配置文件 ros.xml 配置。

密码输入证书实际的密码。

eSight V300R006C00 Open API 用户指南

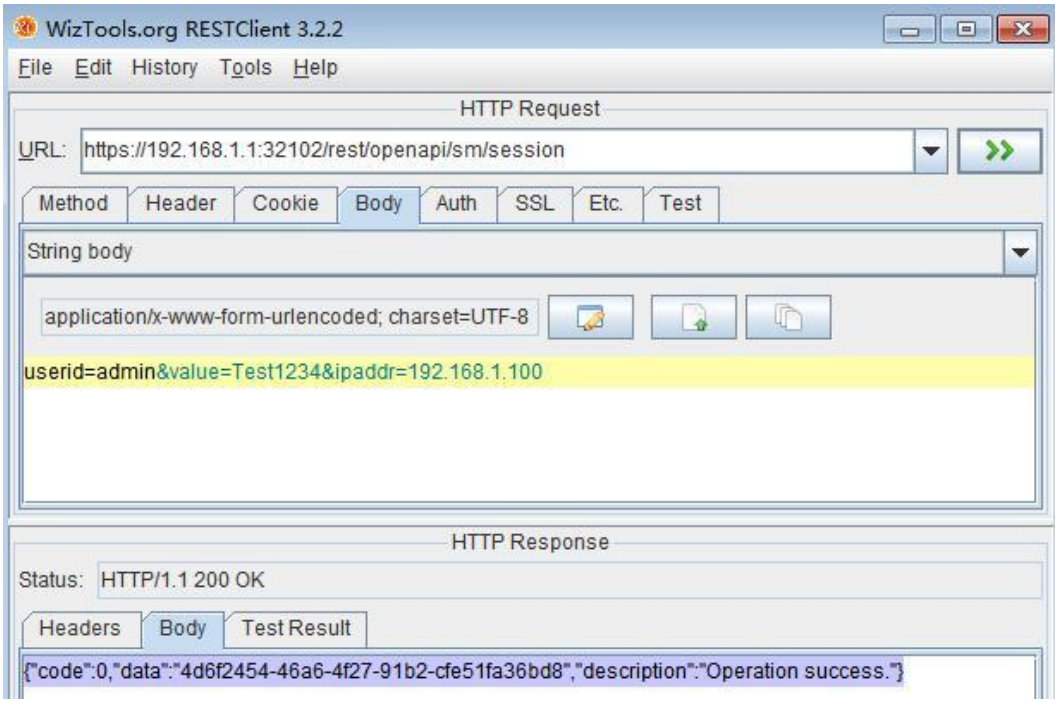


步骤 5 在 SSL 页签的 General 标签中，设置如下。



步骤 6 单击右边的按钮  。

eSight V300R006C00 Open API 用户指南



报文中的“4d6f2454-46a6-4f27-91b2-cfe51fa36bd8”就是 openid。

----结束

调用 Open API 功能接口

说明

当 Method 页签中选择的方法为 DELETE 时，请将参数位置为“参数列表”的参数放在 URL 中，如 http://192.168.1.1:32102/rest/openapi/user?userid=user01，否则接口会调用失败。

这里以查询设备信息的接口为例，其他接口的调用方法类似。

阅读查询设备信息的接口信息，可以获取到如下信息：

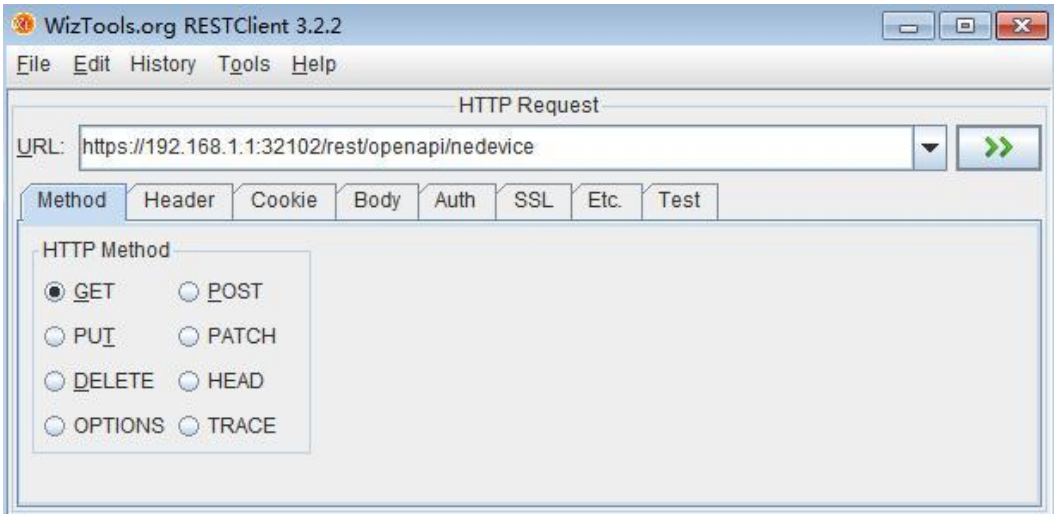
- URL：/rest/openapi/nedevice
- http 方法：GET
- 部分参数信息如下：

参数名	必选/可选	参数位置	参数类型	参数说明
necategory	可选	参数列表	String	设备分类。
netype	可选	参数列表	String	设备类型。
neip	可选	参数列表	String	设备 IP 地址。

步骤 1 输入 URL：https://192.168.1.1:32102/rest/openapi/nedevice。

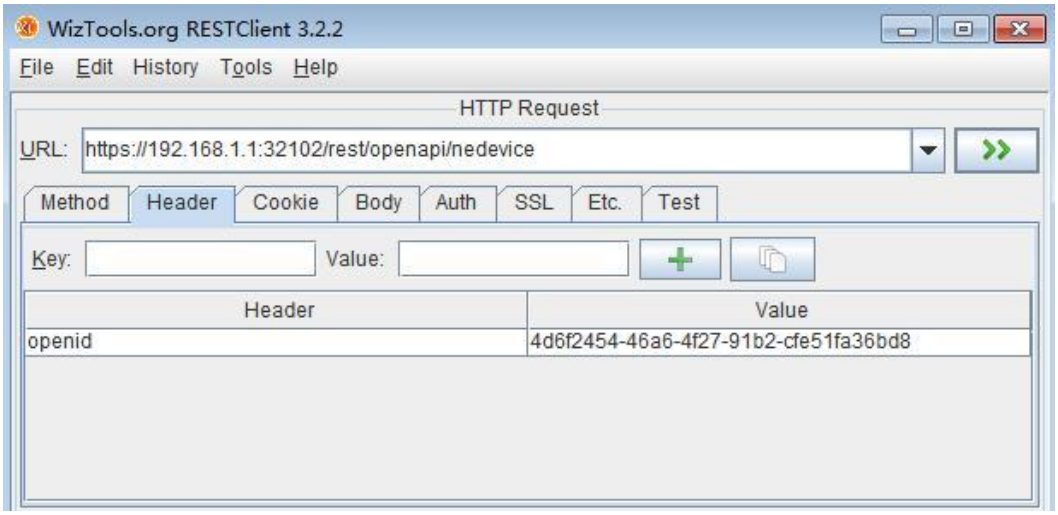
步骤 2 在 Method 页签中选择“GET”。

eSight V300R006C00 Open API 用户指南



步骤 3　在 Headers 页签中添加参数 openid。

key 为“openid”，value 的值就是上面获取的 openid 值“4d6f2454-46a6-4f27-91b2-cfe51fa36bd8”。

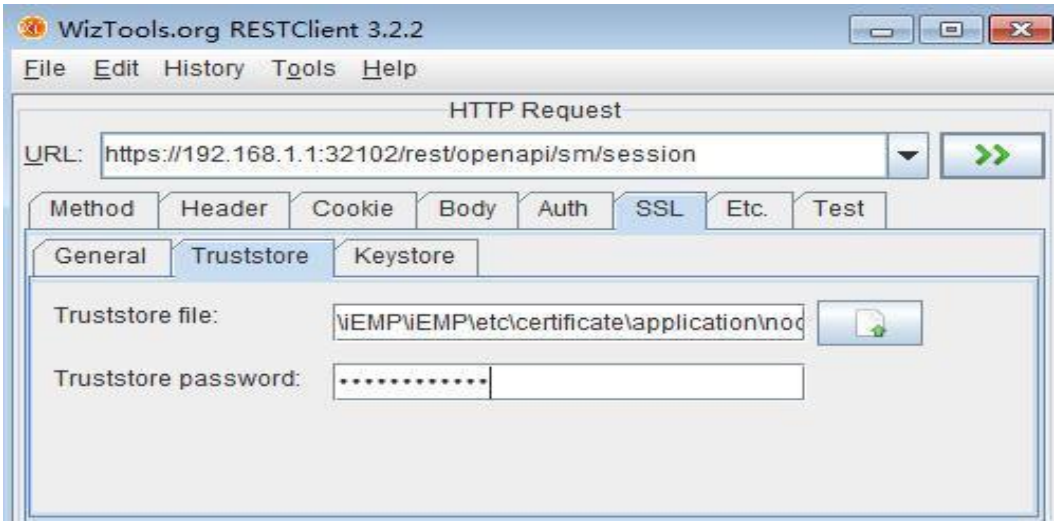


步骤 4　在 SSL 页签的 Truststore 标签中输入证书路径和密码。

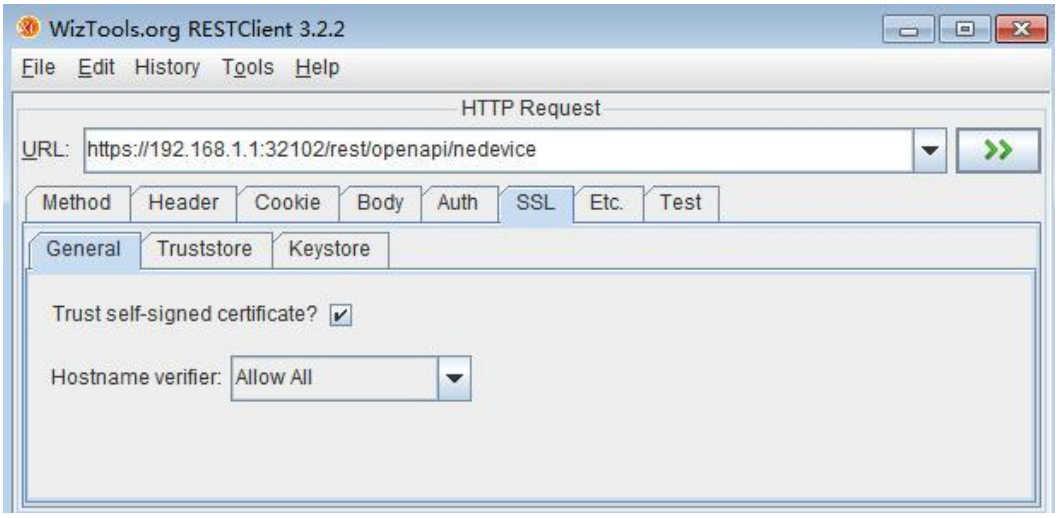
证书可以从网管获取，路径可以参考配置文件 ros.xml 配置。

密码输入证书实际的密码。

eSight V300R006C00 Open API 用户指南

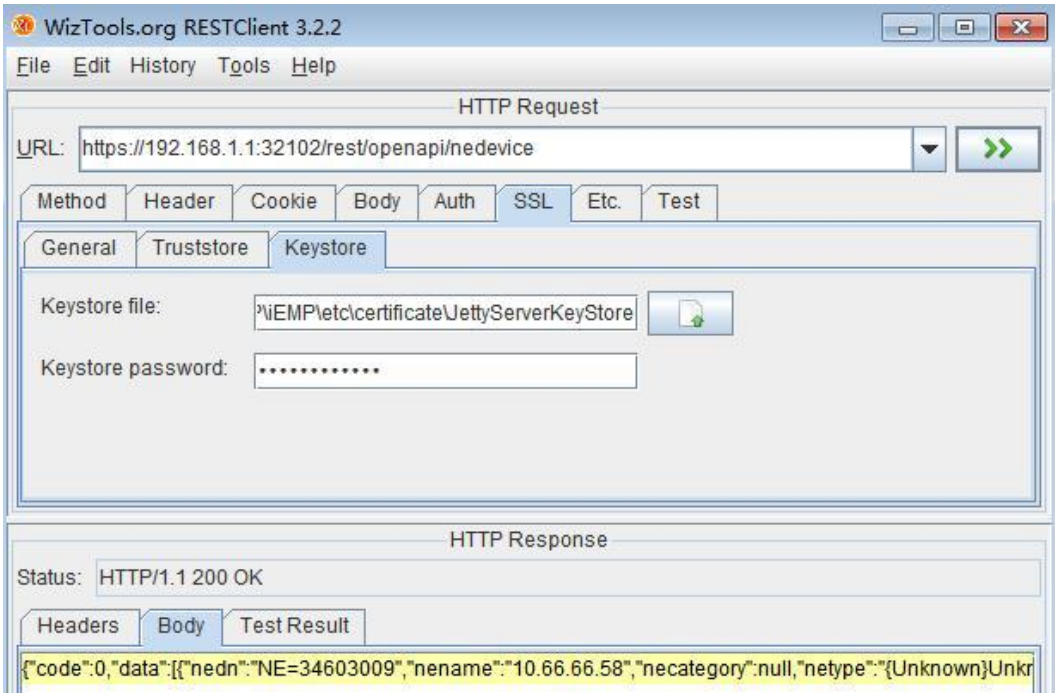


步骤 5 在 SSL 页签的 General 标签中，设置如下。



步骤 6 单击右边的按钮  。

eSight V300R006C00 Open API 用户指南



----结束

3.3 接口调用时的请求与响应示例

本节展示接口调用过程中的请求头和响应信息。

以“查询指定告警流水号的告警信息”为例，请求头和响应分别如下：

请求头：

Request URL: https://10.67.152.137:32102/rest/openapi/alarm
Request Method: GET
Status Code: 200 OK

响应：

```
HTTP/1.1 200 OK [Content-Type: application/json;charset=UTF-8, Content-Length: 91]
```

一次完整的请求如下：

Remote Address: 10.67.152.137:32102
Request URL: https://10.67.152.137:32102/rest/openapi/alarm
Request Method: GET
Status Code: 200 OK
Request Headers view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
Connection: keep-alive
Host: 10.67.152.137:32102
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.153 Safari/537.36
Response Headers view source
Content-Length: 56
Content-Type: application/json;charset=UTF-8

其他接口的请求头与响应信息与此类似。

4 基于 HttpClient 的接口参考

本节介绍基于 HttpClient 的接口参考，参考中的示例均为基于 openid 认证的方式。

说明

- 在调用 Open API 接口时，除“登录 eSight”接口、“退出 eSight”接口外，其他接口都需要先授予“调用 Open API 接口”的权限，否则会返回认证错误，无法调用接口。
- 为确保安全，产品开发的 Open API 接口必须鉴权。

4.1 定位北向管理接口参考

本节介绍定位北向管理接口。

4.1.1 查询区域信息接口

接口功能

查询当前定位服务器中的区域信息。

URL 路径

/rest/openapi/rtls/locationmap

访问方法

GET

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参数列表	String	会话标识，用于 Open API 的鉴权。 说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。
version	必选	参数列表	String	接口版本信息，为固定值： “1.0”
param	可选	参数列表	String	使用的参数，为 json 格式。 如： {"mapID":1} 如果不设置该参数，默认查询所有区域信息

param 包含以下字段：

字段名	类型	说明
mapID	int	区域 ID

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	String	查询得到的信息，json 格式。
description	String	接口调用结果的描述信息。

data 包含以下字段：

字段名	类型	说明
mapId	int	区域 ID（例 333）
mapName	String	区域名称（例 N3）
regionPath	int[]	区域路径 ID，从第一级区域到本区域（例[111,222,333]）
regionPathName	String[]	区域路径名称，从第一级区域到本区域（例[China,Shanghai,N3]）
width	int	区域宽
height	int	区域高
scale	String	区域比例尺
aps	list<ApInfo>	区域中 AP 的列表
obstacles	list<ObstacleInfo>	区域中障碍物的列表
coverages	list<CoverageInfo>	区域中覆盖区域的列表

ApInfo 包含以下字段

字段名	类型	说明
mac	String	AP 的 mac 地址
x	float	AP 的横坐标
y	float	AP 的纵坐标

ObstacleInfo 包含以下字段

字段名	类型	说明
obstacleID	String	障碍物的 ID
attenuation	int	障碍物衰减值
thickness	float	障碍物厚度值
shape	int	障碍物形状
width	float	障碍物宽
height	float	障碍物高
points	list<Point>	障碍物位置信息列表

CoverageInfo 包含以下字段

字段名	类型	说明
id	int	覆盖区域 ID
name	String	覆盖区域名称

注意事项

无。

使用示例

```
import org.apache.http.HttpResponse;
import org.apache.http.message.BasicNameValuePair;

public class Query
{
   private static final String EAM_TYPE = "/rest/openapi/rtls/locationmap";

   public static void main(String[] args)
      throws Exception
   {
      Login.login();
      queryTest();
   }

   public static void queryTest()
      throws Exception
   {
      //set the URL and method
      String openidURL = EAM_TYPE;
      String method = "GET";

      //set headers
      BasicNameValuePair[] headers = NewRosSecurity.getRosHttpHeader(openidURL,
method);
      //set parameters
      BasicNameValuePair[] parameters =
         {
            new BasicNameValuePair("version", "1.0"),
            new BasicNameValuePair("param", "{\"mapID\":3}"),
         };
      //send the request
      HttpResponse response =
         NewHttpsAccess.access(GlobalVar.GLOBAL_IP, GlobalVar.GLOBAL_PORT,
openidURL, method, headers, parameters);
      //get the result
      String body = NewHttpsAccess.getResult(response);
      System.out.println(body);
```

```
    }
}
执行结果如下：
{
"code":0,"data":[{"mapId":3,"mapName":"3F","width":2407,"height":779,"scale":"0.025
","regionPath":[1,2,3],"regionPathName":["
nanjing","N3","3F"],"aps":[{"mac":"C7-A4-01-01-01-
01","x":1167.0,"y":451.0},{"mac":"C7-A4-01-01-01-
02","x":628.0,"y":525.0},{"mac":"C7-A4-01-01-01-
03","x":165.0,"y":301.0},{"mac":"C7-A4-01-01-01-
04","x":626.0,"y":194.0},{"mac":"C7-A4-01-01-01-
05","x":317.0,"y":452.0},{"mac":"C7-A4-01-01-01-
06","x":1000.0,"y":116.0}],"obstacles":[],"coverages":[{"id":4,"name":"B"},{"id":5,
"name":"A"},{"id":6,"name":"D"}]}],
"description":"Operation success."
}
```

4.1.2 查询终端定位信息接口

接口功能

查询当前定位服务器中的终端定位信息。

URL 路径

/rest/openapi/rtls/locationquery

访问方法

GET

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参数列表	String	会话标识，用于 Open API 的鉴权。 说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。
version	必选	参数列表	String	接口版本信息，为固定值： “1.0”
param	可选	参数列表	String	使用的参数，为 json 格式。 如： {"starttime":1435023510052,"endtime":1435048162874} 如果不设置该参数，默认查询实

参数名	必选/可选	参数位置	参数类型	参数说明
				时终端定位信息

param 包含以下字段：

字段名	类型	说明
starttime	long	需要查询的起始时间，自 1970 年 1 月 1 号 0 时刻到当前时间点的时间（毫秒数），单位：ms
endtime	long	需要查询的结束时间，自 1970 年 1 月 1 号 0 时刻到当前时间点的时间（毫秒数），单位：ms

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	查询得到终端的定位信息列表。
description	String	接口调用结果的描述信息。

data 中的每一个信息包含以下字段：

字段名	类型	说明
idtype	String	userid 的类型，为空
userid	String	终端的唯一标识
datatype	String	数据类型，只能是 coordinates
apMac	String	信号最强 AP 的 MAC 地址 (非实时查询时为空)
apRssi	String	信号最强 AP 的 RSSI 值 (非实时查询时为空)
associationState	String	"0"：未关联; "1"：关联; (非实时查询时为空)
terminalIp	String	终端 IP 地址,为空 (非实时查询时为空)

字段名	类型	说明
location	LocationData	定位信息数据
timestamp	long	定位数据的时间戳，自 1970 年 1 月 1 号 0 时刻到当前时间点的时间（毫秒数），单位：ms

LocationData 包含以下字段

字段名	类型	说明
x	float	终端的横坐标
y	float	终端的纵坐标
z	int	终端所在的 map 的 ID

注意事项

无。

使用示例

```
import org.apache.http.HttpResponse;
import org.apache.http.message.BasicNameValuePair;

public class QueryLocation
{
   private static final String EAM_TYPE = "/rest/openapi/rtls/locationquery";

   public static void main(String[] args)
      throws Exception
   {
      Login.login();
      queryLocationTest();
   }

   public static void queryLocationTest()
      throws Exception
   {
      //set the URL and method
      String openidURL = EAM_TYPE;
      String method = "GET";

      //set headers
      BasicNameValuePair[] headers = NewRosSecurity.getRosHttpHeader(openidURL,
method);
      //set parameters
      BasicNameValuePair[] parameters =
         {
```

eSight V300R006C00 Open API 用户指南


```
            new BasicNameValuePair("version", "1.0"),
            new BasicNameValuePair("param",
"{\"starttime\":1435023510052,\"endtime\":1435048162874}"),
         };
      //send the request
      HttpResponse response =
         NewHttpsAccess.access(GlobalVar.GLOBAL_IP, GlobalVar.GLOBAL_PORT,
openidURL, method, headers, parameters);
      //get the result
      String body = NewHttpsAccess.getResult(response);
      System.out.println(body);
   }
}
```

执行结果如下：

```
{
"code":0,
"data":[{"idtype":"","userid":"203413968453890","datatype":"coordinates","location"
:{"x":375.0,"y":395.0,"z":1},"timestamp":"1435023510052"},{"idtype":"","userid":"20
3413968453890","datatype":"coordinates","location":{"x":375.0,"y":398.0,"z":1},"tim
estamp":"1435023525053"},{"idtype":"","userid":"203413968453890","datatype":"coordi
nates","location":{"x":375.0,"y":401.0,"z":1},"timestamp":"1435023540053"},{"idtype
":"","userid":"203413968453890","datatype":"coordinates","location":{"x":375.0,"y":
404.0,"z":1},"timestamp":"1435023555052"}],
"description":"Operation success."
}
```

eSight V300R006C00 Open API 用户指南

5 消息主动通知参考

eSight 通过 HTTP/HTTPS 协议，向第三方系统发送订阅过的资源变更信息，包括资源的新增、删除、修改。

5.1 终端区域进出消息

5.1.1 订阅终端区域进出通知

接口功能

订阅终端区域进出消息通知接口。

URI 路径

/rest/openapi/notification/rtls/regioninout

访问方法

PUT

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参数列表	String	会话标识，用于 Open API 的鉴权。

eSight V300R006C00 Open API 用户指南

参数名	必选/可选	参数位置	参数类型	参数说明
				说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。
systemID	必选	请求实体参数列表	String	第三方系统标识。 说明 可为 IP 地址，1~64 个字符，字符集合为英文半角：0-9a-zA-Z@_-().,^$~`!
openID	必选	请求实体参数列表	String	网管主动连接第三方系统的认证凭证，由第三方系统分配和利用此字符串认证。 说明 1~64 个字符，字符集同 systemID。
url	必选	请求实体参数列表	String	网管以 POST 方式向该 URL 发送通知消息。 说明 例如 http://10.10.10.10:8080/device。需要通知消息订阅者确保 URL 的正确性，IP 地址为 OpenAPI 白名单列表中地址。长度 1~1024，字符串符合 HTTP URL 编码规范。
dataType	可选	请求实体参数列表	String	通知报文的 data 字段类型。 说明 目前仅支持“JSON”，缺省为“JSON”。
desc	可选	请求实体参数列表	String	第三方系统描述。 说明 缺省为 null 不设置。如设置长度限制 0~1024 字符，字符集合不限制。

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null

字段名	类型	说明
description	String	接口调用结果的描述信息。

注意事项

无。

5.1.2 取消订阅终端区域进出通知

接口功能

取消订阅终端区域进出消息通知的接口。

URI 路径

/rest/openapi/notification/rtls/regioninout

访问方法

DELETE

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
systemID	必选	参数列表	String	第三方系统标识（可为 IP 地址）
desc	可选	参数列表	String	第三方系统描述

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.1.3 终端区域进出通知数据结构

消息数据

字段名	类型	说明
resourceURI	String	资源类型 URI，与订阅消息的 URI 定义一致，该变更通知固定为“/rest/openapi/notification/rtls/regioninout”字符串。
msgType	int	消息类型。取值范围： • 3：修改
data	List	业务数据
description	String	描述信息
timestamp	String	事件发生的时间，eSight 服务器所在时区的协调时间; 格式：yyyy-mm-dd hh:MM:ss

data 每一个信息包含以下字段：

字段名	类型	说明
idtype	String	userid 的类型，为空
userid	String	终端的唯一标识
timestamp	long	进入/离开区域的时间，自 1970 年 1 月 1 号 0 时刻到当前时间点的时间（毫秒数），单位：ms
regionId	int	进出区域 id
regionType	int	1：覆盖区域，2：底层区域，3：非底层区域
inoutType	int	0：离开区域，1：进入区域

通知示例

消息通知以 POST 方式提交，所有通知数据都保存在 HTTP 报文 Body 中。假设第三方系统 WebService 提供 javax.servlet.http.HttpServletRequest 的实现 request 实例（其他语言也有类似库实现），可如下获取消息数据：

调用方法	取得的值

调用方法	取得的值
request.getParameter("resourceURI")	"/rest/openapi/notification/rtls/regioninout"
request.getParameter("msgType")	3
request.getParameter("data")	"[{ \"idtype\": \"\", \"userid": \"1111\", \"timestamp\": \"1445952206033\", \"regionId\": 34603109, \"regionType\": 3, \"inoutType\": 1 }]"
request.getParameter("description")	"Terminal enters or leaves region notification"
request.getParameter("timestamp")	"2015-10-28 16:29:31"

说明
返回值为“null”时，表示属性实际值为空。

5.2 终端最强 AP 消息

5.2.1 订阅终端最强 AP 通知

接口功能

1. 订阅终端最强 AP 消息通知接口
2. 终端进入定位区域通知（含离开后再次进入）
3. 每分钟推送一次所有终端的最强 AP 定位数据

URI 路径

/rest/openapi/notification/rtls/strongestap

访问方法

PUT

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参	String	会话标识，用于 Open API 的鉴

参数名	必选/可选	参数位置	参数类型	参数说明
		数列表		权。 说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。
systemID	必选	请求实体参数列表	String	第三方系统标识。 说明 可为 IP 地址，1~64 个字符，字符集合为英文半角：0-9a-zA-Z@_-(),.^$~`!
openID	必选	请求实体参数列表	String	网管主动连接第三方系统的认证凭证，由第三方系统分配和利用此字符串认证。 说明 1~64 个字符，字符集同 systemID。
url	必选	请求实体参数列表	String	网管以 POST 方式向该 URL 发送通知消息。 说明 例如 http://10.10.10.10:8080/device。需要通知消息订阅者确保 URL 的正确性，IP 地址为 OpenAPI 白名单列表中地址。长度 1~1024，字符串符合 HTTP URL 编码规范。
dataType	可选	请求实体参数列表	String	通知报文的 data 字段类型。 说明 目前仅支持“JSON”，缺省为“JSON”。
desc	可选	请求实体参数列表	String	第三方系统描述。 说明 缺省为 null 不设置。如设置长度限制 0~1024 字符，字符集合不限制。

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败

字段名	类型	说明
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.2.2 取消订阅终端最强 AP 通知

接口功能

取消订阅终端最强 AP 消息通知的接口。

URI 路径

/rest/openapi/notification/rtls/strongestap

访问方法

DELETE

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
systemID	必选	参数列表	String	第三方系统标识（可为 IP 地址）
desc	可选	参数列表	String	第三方系统描述

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.2.3 终端最强 AP 通知数据结构

消息数据

字段名	类型	说明
resourceURI	String	资源类型 URI，与订阅消息的 URI 定义一致，该变更通知固定为“/rest/openapi/notification/rtls/strongestap”字符串。
msgType	int	消息类型。取值范围： • 1：新增 （终端进入区域被定位到） • 3：修改
data	List	业务数据
description	String	描述信息
timestamp	String	事件发生的时间，eSight 服务器所在时区的协调时间; 格式：yyyy-mm-dd hh:MM:ss

data 每一个信息包含以下字段：

字段名	类型	说明
terminalId	long	终端 ID
type	String	固定为“STA”
terminalInfo	String	空
timeStamp	long	定位时间，自 1970 年 1 月 1 号 0 时刻到当前时间点的时间（毫秒数），单位：ms
strongestAPMac	String	最强 AP 的 MAC

通知示例

消息通知以 POST 方式提交，所有通知数据都保存在 HTTP 报文 Body 中。假设第三方系统 WebService 提供 javax.servlet.http.HttpServletRequest 的实现 request 实例（其他语言也有类似库实现），可如下获取消息数据：

调用方法	取得的值
request.getParameter("resourceURI	"/rest/openapi/notification/rtls/strongestap"

调用方法	取得的值
")	
request.getParameter("msgType")	3
request.getParameter("data")	"[{ \"terminalId\": 123, \"type\": \"STA\", \"terminalInfo\":\"\" \"timestamp\": \"1445952206033\", \"strongestAPMac\": \"AA-BB-CC-DD-EEFF\" }]"
request.getParameter("description")	"The strongest AP location notification"
request.getParameter("timestamp")	"2015-10-30 14:13:49"

说明
返回值为“null”时，表示属性实际值为空。

5.3 进出店变更消息

5.3.1 订阅进出店变更消息

接口功能

订阅进出店变更消息通知接口。

URI 路径

/rest/openapi/notification/rtls/regionalaccess

访问方法

PUT

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参数列表	String	会话标识，用于 Open API 的鉴权。 说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。

参数名	必选/可选	参数位置	参数类型	参数说明
systemID	必选	请求实体参数列表	String	第三方系统标识。 说明 可为 IP 地址，1~64 个字符，字符集合为英文半角：0-9a-zA-Z@_-(),.^$~`!
openID	必选	请求实体参数列表	String	网管主动连接第三方系统的认证凭证，由第三方系统分配和利用此字符串认证。 说明 1~64 个字符，字符集同 systemID。
url	必选	请求实体参数列表	String	网管以 POST 方式向该 URL 发送通知消息。 说明 例如 http://10.10.10.10:8080/device。需要通知消息订阅者确保 URL 的正确性，IP 地址为 OpenAPI 白名单列表中地址。长度 1~1024，字符串符合 HTTP URL 编码规范。
dataType	可选	请求实体参数列表	String	通知报文的 data 字段类型。 说明 目前仅支持“JSON”，缺省为“JSON”。
desc	可选	请求实体参数列表	String	第三方系统描述。 说明 缺省为 null 不设置。如设置长度限制 0~1024 字符，字符集合不限制。

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.3.2 取消订阅进出店变更消息

接口功能

取消订阅进出店变更消息通知接口。

URI 路径

/rest/openapi/notification/rtls/regionalaccess

访问方法

DELETE

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
systemID	必选	参数列表	String	第三方系统标识（可为 IP 地址）
desc	可选	参数列表	String	第三方系统描述

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.3.3 进出店变更消息数据结构

消息数据

字段名	类型	说明
resourceURI	String	资源类型 URI，与订阅消息的 URI 定义一致，该变更通知固定为“/rest/openapi/notification/rtls/regionalaccess”字符串。
msgType	int	消息类型。取值范围： • 3：修改
data	List	业务数据
description	String	描述信息
timestamp	String	事件发生的时间，eSight 服务器所在时区的协调时间; 格式：yyyy-mm-dd hh:MM:ss

data 每一个信息包含以下字段：

字段名	类型	说明
idtype	String	标识 terminalId 的类型，值为"ID"
terminalid	String	进出店终端的唯一标识
bz	int	1 表示进店，3 表示出店
regionid	int	区域 id
regioninfo	String	区域信息，为店铺内 AP 的 MAC 地址
timestamp	String	进出店时间戳

通知示例

消息通知以 POST 方式提交，所有通知数据都保存在 HTTP 报文 Body 中。假设第三方系统 WebService 提供 javax.servlet.http.HttpServletRequest 的实现 request 实例（其他语言也有类似库实现），可如下获取消息数据：

调用方法	取得的值
request.getParameter("resourceURI")	"/rest/openapi/notification/rtls/regionalaccess"
request.getParameter("msgType")	3
request.getParameter("data")	"[{

调用方法	取得的值
	"idtype":"ID", "terminalid":"1111", "bz":1, "regionid":1, "regioninfo":"A5-01-15-0B-01-07", "timestamp":1459258602180 }]"
request.getParameter("description")	"Terminal enters or leaves coverage region notification"
request.getParameter("timestamp")	"2016-03-29 21:40:16"

说明
返回值为“null”时，表示属性实际值为空。

5.4 实时终端定位推送消息

5.4.1 订阅实时终端定位消息

接口功能

订阅实时终端定位通知接口。

URI 路径

/rest/openapi/notification/rtls/realtimelocatedata

访问方法

PUT

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参数列表	String	会话标识，用于 Open API 的鉴权。 说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。
systemID	必选	请求实体参数列表	String	第三方系统标识。 说明

参数名	必选/可选	参数位置	参数类型	参数说明
				可为 IP 地址，1~64 个字符，字符集合为英文半角：0-9a-zA-Z@_-(),.^$~`!
openID	必选	请求实体参数列表	String	网管主动连接第三方系统的认证凭证，由第三方系统分配和利用此字符串认证。 说明 1~64 个字符，字符集同 systemID。
url	必选	请求实体参数列表	String	网管以 POST 方式向该 URL 发送通知消息。 说明 例如 http://10.10.10.10:8080/device。需要通知消息订阅者确保 URL 的正确性，IP 地址为 OpenAPI 白名单列表中地址。长度 1~1024，字符串符合 HTTP URL 编码规范。
conditon	可选	请求实体参数列表	String	可选参数，支持通知间隔和终端过滤条件设置，不同的站点订阅时可设置不同的通知间隔和终端过滤条件。 通知间隔单位为秒，默认为 15s，参数为 json 格式。如： {"dataSendInterval":20} 终端过滤条件通过设置指定的终端 ID，实现不同的站点订阅不同的终端信息，默认为订阅所有的终端信息，参数为 json 格式。如： {"terminalId":" 2,3"} 通知间隔和终端过滤条件支持同时设置。如： {"dataSendInterval":20,"terminalId":"2,3"}
domainSite	可选	请求实体参数列表	String	产品订阅时的站点域标志，当有多个不同站点订阅时，必须输入，用以区分不同的站点。
dataType	可选	请求实体参数列表	String	通知报文的 data 字段类型。 说明 目前仅支持“JSON”，缺省为“JSON”。

参数名	必选/可选	参数位置	参数类型	参数说明
desc	可选	请求实体参数列表	String	第三方系统描述。 说明 缺省为 null 不设置。如设置长度限制 0~1024 字符，字符集合不限制。

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.4.2 取消订阅实时终端定位消息

接口功能

取消订阅实时终端定位通知的接口。

URI 路径

/rest/openapi/notification/rtls/realtimelocatedata

访问方法

DELETE

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
systemID	必选	参数列表	String	第三方系统标识（可为 IP 地址）
domainSite	可选	参数列表	String	产品订阅时的站点域标志

参数名	必选/可选	参数位置	参数类型	参数说明
desc	可选	参数列表	String	第三方系统描述

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.4.3 实时终端定位推送消息数据结构

消息数据

字段名	类型	说明
resourceURI	String	资源类型 URI，与订阅消息的 URI 定义一致，该变更通知固定为“/rest/openapi/notification/rtls/realtimelocatedata”字符串。
msgType	int	消息类型。取值范围： • 3：修改
data	List	业务数据
description	String	描述信息
timestamp	String	事件发生的时间，eSight 服务器所在时区的协调时间; 格式：yyyy-mm-dd hh:MM:ss

data 每一个信息包含以下字段：

字段名	类型	说明

字段名	类型	说明
idtype	String	terminalId 的类型，值为"ID"
terminalId	String	终端的唯一标识
unit	String	坐标的单位值，固定值"METERS"
x	String	X 轴坐标
y	String	Y 轴坐标
mapId	String	定位到地图的 ID
mapName	String	定位所在地图的名称
apMac	String	信号最强 AP 的 MAC 地址
apRssi	String	信号最强 AP 接收的 RSSI 值
timestamp	String	探测到该终端时的时间戳

通知示例

消息通知以 POST 方式提交，所有通知数据都保存在 HTTP 报文 Body 中。假设第三方系统 WebService 提供 javax.servlet.http.HttpServletRequest 的实现 request 实例（其他语言也有类似库实现），可如下获取消息数据：

调用方法	取得的值
request.getParameter("resourceURI")	"/rest/openapi/notification/rtls/realtimelocatedata"
request.getParameter("msgType")	3
request.getParameter("data")	"[{ "idtype":"ID", "terminalId":"2", "unit":"METERS" ,"x":"12.192", "y":"-1.9558", "mapId":"34603011", "mapName":"test_hh", "apMac":"C7-A4-01-01-01-10", "apRssi":"-22", "timestamp":"1459242063206" }]"
request.getParameter("description")	"Real Time Location Data"
request.getParameter("timestamp")	"2016-05-09 17:01:16"

说明
返回值为“null”时，表示属性实际值为空。

5.5 终端定位信息变更消息

5.5.1 订阅终端定位信息变更消息

接口功能

1. 订阅终端定位信息变更主动通知
2. 支持设置通知间隔

URI 路径

/rest/openapi/notification/rtls/terminallocinfo

访问方法

PUT

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
openid	必选	请求头参数列表	String	会话标识，用于 Open API 的鉴权。 说明 该参数由第三方调用安全管理**错误！未找到引用源。错误！未找到引用源。**获取 openid。
systemID	必选	请求实体参数列表	String	第三方系统标识。 说明 可为 IP 地址，1~64 个字符，字符集合为英文半角：0-9a-zA-Z@_-().,^$~`!
openID	必选	请求实体参数列表	String	网管主动连接第三方系统的认证凭证，由第三方系统分配和利用此字符串认证。 说明 1~64 个字符，字符集同 systemID。
url	必选	请求实体参数列表	String	网管以 POST 方式向该 URL 发送通知消息。 说明

参数名	必选/可选	参数位置	参数类型	参数说明
				例如 http://10.10.10.10:8080/device。需要通知消息订阅者确保 URL 的正确性，IP 地址为 OpenAPI 白名单列表中地址。长度 1~1024，字符串符合 HTTP URL 编码规范。
conditon	可选	请求实体参数列表	String	可选参数，支持通知间隔设置，不同的站点订阅时可设置不同的通知间隔。单位为秒，默认为 15s,参数为 json 格式。如： {"dataSendInterval":20}
domainSite	可选	请求实体参数列表	String	产品订阅时的站点域标志，当有多个不同站点订阅时，必须输入，用以区分不同的站点。
dataType	可选	请求实体参数列表	String	通知报文的 data 字段类型。 说明 目前仅支持“JSON”，缺省为“JSON”。
desc	可选	请求实体参数列表	String	第三方系统描述。 说明 缺省为 null 不设置。如设置长度限制 0~1024 字符，字符集合不限制。

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

eSight V300R006C00 Open API 用户指南

5.5.2 取消订阅终端定位信息变更消息

接口功能

取消订阅终端定位信息变更消息通知的接口。

URI 路径

/rest/openapi/notification/rtls/terminallocinfo

访问方法

DELETE

参数说明

参数名	必选/可选	参数位置	参数类型	参数说明
systemID	必选	参数列表	String	第三方系统标识（可为 IP 地址）
domainSite	可选	参数列表	String	产品订阅时的站点域标志
desc	可选	参数列表	String	第三方系统描述

返回结果

字段名	类型	说明
code	int	操作返回码。可以是如下值之一： • 0：成功 • 非 0：失败
data	List	此处为 null
description	String	接口调用结果的描述信息。

注意事项

无。

5.5.3 终端定位信息变更消息数据结构

消息数据

字段名	类型	说明

字段名	类型	说明
resourceURI	String	资源类型 URI，与订阅消息的 URI 定义一致，该变更通知固定为“/rest/openapi/notification/rtls/terminallocinfo”字符串。
msgType	int	消息类型。取值范围： • 3：修改
data	List	业务数据
description	String	描述信息
timestamp	String	事件发生的时间，eSight 服务器所在时区的协调时间; 格式：yyyy-mm-dd hh:MM:ss

data 每一个信息包含以下字段：

字段名	类型	说明
idtype	String	terminalId 的类型，为空
terminalId	String	终端的唯一标识
timestamp	String	最近一次探测到该终端时的时间戳，自 1970 年 1 月 1 号 0 时刻到当前时间点的时间（毫秒数），单位：ms
apMac	String	信号最强 AP 的 MAC 地址
apRssi	String	信号最强 AP 的 RSSI 值
associationState	String	"0"：未关联; "1"：关联;
terminalIp	String	终端 IP 地址,为空
locType	String	"1"：新增; "2"：删除; "3"：更新;

通知示例

消息通知以 POST 方式提交，所有通知数据都保存在 HTTP 报文 Body 中。假设第三方系统 WebService 提供 javax.servlet.http.HttpServletRequest 的实现 request 实例（其他语言也有类似库实现），可如下获取消息数据：

调用方法	取得的值
request.getParameter("resourceURI")	"/rest/openapi/notification/rtls/terminallocinfo"

调用方法	取得的值
request.getParameter("msgType")	3
request.getParameter("data")	"[{ \"idtype\": \"\", \"terminalId\": \"1111\", \"timestamp\": \"1445952206033\", \"apMac\": \"80-71-7A-FF-1D-1A\", \"apRssi\": \"-70\", \"associationState\": \"0\", \"teminalIp\":\"\", \"locType\":\"1\" }]"
request.getParameter("description")	"Terminal location information change notification"
request.getParameter("timestamp")	"2016-05-10 16:29:31"

说明
返回值为“null”时，表示属性实际值为空。

eSight V300R006C00 Open API 用户指南

6 FAQ

介绍常见问题和参考信息，让您在问题出现时能采取合适的应对处理策略。

6.1 如何解决连接被拒绝问题？

现象描述

调用 Open API 时出现如下错误，请问如何解决？

```
Exception in thread "main" org.apache.http.conn.HttpHostConnectException:
Connection to https://10.66.66.58:32102 refused
 at
org.apache.http.impl.conn.DefaultClientConnectionOperator.openConnection(DefaultCli
entConnectionOperator.java:190)
 at
org.apache.http.impl.conn.ManagedClientConnectionImpl.open(ManagedClientConnectionI
mpl.java:294)
 at
org.apache.http.impl.client.DefaultRequestDirector.tryConnect(DefaultRequestDirecto
r.java:640)
 at
org.apache.http.impl.client.DefaultRequestDirector.execute(DefaultRequestDirector.j
ava:479)
 at
org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:906)
 at
org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:805)
 at
org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:784)
 at com.huawei.oms.ros.test.NewHttpsAccess.access(NewHttpsAccess.java:155)
 at com.huawei.oms.ros.test.Login.login(Login.java:34)
 at com.huawei.oms.ros.test.QueryLogs.main(QueryLogs.java:17)
Caused by: java.net.ConnectException: Connection refused: connect
 at java.net.DualStackPlainSocketImpl.connect0(Native Method)
 at
```

```
java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:69)
 at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:339)
 at
java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:200)
 at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:182)
 at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:157)
 at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:391)
 at java.net.Socket.connect(Socket.java:579)
 at sun.security.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:612)
 at
org.apache.http.conn.ssl.SSLSocketFactory.connectSocket(SSLSocketFactory.java:549)
 at
org.apache.http.impl.conn.DefaultClientConnectionOperator.openConnection(DefaultCli
entConnectionOperator.java:180)
 ... 9 more
```

可能原因

- 服务器的 Open API 端口（缺省为 32102）没有处于侦听状态。

说明
企业解决方案指定 Open API 访问端口为 32102，请不要使用 8086、31943、32101 这些端口来访问 Open API。

- 配置文件 ros.xml 的“ip”项或“keystorePass”项配置错误。

处理步骤

步骤 1 检查 Open API 端口 32102 是否处于侦听状态。Windows 和 Linux、Solaris 操作系统中命令分别如下。

- Windows：**netstat -an|findstr** 32102
- Linux、Solaris：**netstat -an|grep** 32102

以 Windows 下为例介绍如何判断端口是否处于侦听状态：

- 如果执行命令后没有返回值，表示端口未处于侦听状态，转步骤 2。
- 如果执行命令后返回“LISTENING”，则表示端口处于侦听状态，转步骤 3。

```
C:\>netstat -an|findstr 32102
TCP    10.10.10.10:32102      0.0.0.0:0              LISTENING
TCP    [::]:32102             [::]:0                 LISTENING
```

步骤 2 检查证书密码和 ros.xml 配置文件中的密码是否一致。

说明
如果端口不处于侦听状态，常见原因是证书（etc/certificate/JettyServerKeyStore）密码和配置文件（etc/oms.ros/ros.xml）中的密码不一致。

1. 确认证书密码正确。

   在 JRE 的 bin 目录下执行命令：

   ```
   keytool -v -list -keystore <安装目录>/iEMP/etc/certificate/JettyServerKeyStore -
   storepass 证书密码
   ```

   例如，以下执行情况表示证书的密码不是 Changeme123@。

   ```
   C:\Java\jdk1.7.0_04\bin>keytool -v -list -keystore
   d:\iemp\iemp/etc/certificate/JettyServerKeyStore -storepass Changeme123@
   ```

eSight V300R006C00 Open API 用户指南


```
keytool 错误： java.io.IOException: Keystore was tampered with, or password was
incorrect
java.io.IOException: Keystore was tampered with, or password was incorrect
       at sun.security.provider.JavaKeyStore.engineLoad(JavaKeyStore.java:771)
       at
sun.security.provider.JavaKeyStore$JKS.engineLoad(JavaKeyStore.java:38)
       at java.security.KeyStore.load(KeyStore.java:1185)
       at sun.security.tools.KeyTool.doCommands(KeyTool.java:620)
       at sun.security.tools.KeyTool.run(KeyTool.java:172)
       at sun.security.tools.KeyTool.main(KeyTool.java:166)
Caused by: java.security.UnrecoverableKeyException: Password verification
failed
       at sun.security.provider.JavaKeyStore.engineLoad(JavaKeyStore.java:769)
       ... 5 more
```

以下执行情况表示证书的密码是 Changeme_123。

```
C:\Java\jdk1.7.0 04\bin>keytool -v -list -keystore
d:\iemp\iemp/etc/certificate/JettyServerKeyStore -storepass Changeme 123

Keystore 类型： JKS
Keystore 提供者： SUN

您的 keystore 包含 1 输入

别名名称： server
创建日期： 2012-12-29
项类型：PrivateKeyEntry
认证链长度： 1
认证 [1]:
所有者:CN=omsMaster, OU=Developer, O=Techstar, L=ShenZhen, ST=ShenZhen, C=CH
签发人:CN=omsMaster, OU=Developer, O=Techstar, L=ShenZhen, ST=ShenZhen, C=CH
序列号:50de6922
有效期：Sat Dec 29 11:53:06 GMT+08:00 2012 至 Tue Dec 27 11:53:06 GMT+08:00 2022
证书指纹:
        MD5:90:DB:D2:38:86:42:D9:56:9A:7F:0F:B5:EE:7B:C3:09
        SHA1:60:5E:E6:33:DD:6F:17:8D:A6:44:4A:E8:80:24:64:3F:DC:13:F3:53
        签名算法名称:SHA1withRSA
        版本：3



*******************************************
*******************************************
```

2. 获取证书密码对应的密文。

   在 <安装目录>/AppBase/tools/bmetool/encrypt 路径下执行 **encrypt.bat 0**，然后根据提示输入新密码。

   执行成功后，输出结果为加密后密文。

   例如，输入 Changeme_123，输出密文为 9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555fd5aff9908d。

3. 将证书密码的密文配置到配置文件（etc/oms.ros/ros.xml）中。

```
<?xml version="1.0" encoding="UTF-8"?>
<config name="oms">
 <config name="webserver">
  <param name="ip">网管服务器实际的 IP 地址</param>
```

eSight V300R006C00 Open API 用户指南


```
 <param name="port">32102</param>
 <param name="keystorefilePath">etc/certificate/JettyServerKeyStore</param>
 <param
name="keystorePass">9d7961bc8af54d05ce509e03b13ffce3abc7587373e7719b62555fd5aff
9908d</param>
 </config>
</config>
```

步骤 3 将配置文件（etc/oms.ros/ros.xml）中“ip”修改为网管服务器实际的 IP 地址。

说明
不能设置为 127.0.0.1。

步骤 4 重启网管。

----结束

6.2 如何解决 License 无效问题？

现象描述

调用 Open API 时出现如下错误，请问如何解决？

```
{"description":"License is invalid.","data":"null","code":1002}
```

可能原因

License 文件不合法。

处理步骤

请参照申请 License 解决。