文档快速开发指南

前言

可以先预览下EnOS 2.0 API 设计指南(draft)以及EnOS 2.0 API 设计指南（2019年Pre-Release），再看本篇的快速开发指南。本篇主要教你怎么写一个open api的文档，从九个组成部分一一介绍。

注意所有要暴露的接口文档必需得通过评审！！！

文档写在哪里？如何写？参考快速上手 API 文档编辑

1 API简介

该部分可以通过一段话简单介绍该接口的功能即可。

示例：Create Asset Tree 接口的简介为——创建一棵资产树，并同时创建该资产树的根节点。

2 操作权限

目前DMG团队的接口涉及两类权限，分别为资产粒度权限和SA粒度权限。想要明白自己的服务属于哪类权限，可参考DMG权限文档（设计说明书：权限管理）并补充新增接口。

示例：Create Asset Tree 接口操作权限为SA粒度，意味着针对的是服务级别的。需要该服务的full access权限才能对该接口进行写操作。

需授权的资源	所需操作权限
资产树管理	Full Access

3 请求格式

如何写该接口的HTTP URL？POST/GET https://{apigw-address} /{serviceName}/{version}/{entity}?}{action}&{parameters}

POST/GET：所有接口（CRUD）必须都支持POST方式，查询类接口还需要支持GET方式，修改类接口不允许支持GET方式；

apigw-address：指的是apim的域名，我们这里原封不动；

serviceName：接口所在服务的服务名称，小写字母，格式为xxx-service，例如asset-service ， asset-tree-service；

version：接口的版本由v开头，点号连接的2个数字组成，第一个数字为产品的大版本号，第二个数字为API的小版本号，例如:v2.0、v2.1

entity：需要操作的资源，小写字母，可通过“-”连接，例如：assets、asset-trees；

action:支持如下几种，可以自定义，采用驼峰式。

- get：单个resource的查询
- search：批量查询
- create：创建
- update：更新
- delete：删除

示例：POST https://{apigw-address}/asset-tree-service/v2.1/asset-trees?action=create

4 请求参数（URI）

请求参数由名称，位置，可选，数据类型，描述组成。

名称：参数名称；

位置：目前按照DMG的URL设计原则，path中不会存在参数，该项都填写Query；

必需可选：参数字段是否必填。必需的参数写在表格的最前面；

数据类型：参数字段的类型

描述：参数字段的描述，有些公共的参数字段，可以通过链接到该参数公共描述位置；

示例：orgId字段的描述在公共的某个位置由专门的详细介绍；

名称	位置（Path/Query）	必需/可选	数据类型	描述
orgId	Query	必需	String	资产所属的组织ID。如何获取orgId信息>>

5 请求参数（Body）

这里基本和4请求参数填写标准一致。除此之外有点非常重要。在创建一个资源时。在body中必需采用平铺的方式，禁止在这些字段基础上再包一层。

例如：创建一个asset：

body

提倡做法：

```
{
 "modelId": "modelName",
 "name": {
    "defaultValue": "Name"
    },
 "timezone": "+12:00",
 "description": "Example of description"
}
```

禁止做法：

```
{
 "asset": {
     "modelId": "modelName",
     "name": {
        "defaultValue": "Name"
        },
     "timezone": "+12:00",
    "description": "Example of description"
     }
}
```

6响应参数

包含名称，数据类型和描述。对于单个资源的更新删除操作，返回的消息体中没有data字段；

示例：

名称	数据类型	描述
data	String	创建的资产树ID。

7 错误码

该部分填写该接口除了公共的错误码外所返回的所有的错误码以及错误信息。并且在后面加上公共错误码有链接

示例：

名称	数据类型	描述
17772	The quota of tree reaches ceiling	资产树数量达到OU上限。

8 示例

填写请求示例的json 返回示例的json。

9Java SDK调用示例

贴上使用sdk的demo，注意要把用户当作小白，sdk要尽可能多的填写你的注释。

示例：

```
import com.alibaba.fastjson.JSON;

import java.util.HashMap;
import java.util.Map;

public class AssetTreeTest {
    private static String AccessKey = "yourAccessKey";
    private static String SecretKey = "yourSecretKey";
    private static String OrgId = "yourOrgId";
    private static String ServerUrl = "yourServerUrl";

    @Test
    public void testCreateTree() {
        TreeCreateVo tree = new TreeCreateVo(); //
        I18nVo treeName = new I18nVo();//
        treeName.setDefaultValue("treeDefaultName");//
        tree.setName(treeName);
        CreateTreeRequest request = new CreateTreeRequest();//
        Map< String, String > i18nValue = new HashMap();
        i18nValue.put("zh_CN", "assetName");
        I18nVo name = new I18nVo();//
        name.setDefaultValue("assetDefaultName");
        name.setI18nValue(i18nValue);
        AssetCreateVo asset = new AssetCreateVo();//
        asset.setModelId("yourModelId");//
        asset.setName(name);//
        asset.setTimezone("+08:00");//
        asset.setDescription("This is a sampled asset");//
        request.setTree(tree);
        request.setAsset(asset);
        request.setOrgId(OrgId);//ID
        CreateTreeResponse response = Poseidon.config(PConfig.init().appKey(AccessKey).appSecret(SecretKey).
debug())
                .url(ServerUrl)
                .getResponse(request, CreateTreeResponse.class);
        System.out.println("create asset tree; " + JSON.toJSONString(response));

    }
}
```

10 expression说明

目前服务对于expression的定义各自为政，很不统一，列举下来。

发现几个问题：

- 文档与接口操作符实际不符，或者文档中字段缺少；
- 字段不统一；
- 操作符不统一；
- 逻辑运算符 and or 说的不清楚，或者没提；

	模型服务	接入服务				资产服务		告警服务					
字段	**Search Thing Model**	**Search Product**	**Search Device**	**Search Sub-Device**	**Search Command**	**Search Asset Tree**	**Search Asset Node**	**Search Alert Severity**	**Search Alert Type**	**Search Alert Content**	**Search Alert Rule**	**Search Active Alerts**	**Search History Alerts**
modelId	in	= in	= in	= in						= in	= in ! and or	in = or	
tags	= exists not exists					= exists	like exist(文档有误 = exists)						
productKey		= in	= in	= in			like =						

productTags		= exists not exists											
productName		like											
deviceAttributes			= exists not exists in	= exists not exists in									
deviceTags			= exists not exists in	= exists not exists in									
deviceKey			= in	= in			like =						
assetId			= in	= in								in = or	in = or
productType			=										
deviceName			like	like									
status			=	=	=								
commandName					like								
createTime					>、= 和 <		=、<=、>=						
assetIds							in =						
modelIds							in =						
rootModelIds							in =						
name							like						
attributes							=(文档没写)						
severityId								= in				in = or	in = or
typeId									= in			in = or	in = or
parentTypeId									= in				
contentId										= in		in = or	in = or
alertTypeId										= in			
ruleId											= in ! or		
measurepointId											= in ! or	in = or	in = or
deviceStatus											= in ! or		
subTypeId												in = or	in = or
eventType												in = or	in = or
eventId												in = or	n = or
deviceStatus												in = or	in = or
assetPath												in = or	in = or
tag												exists not exists	exists not exists

补充：review文档关注点

1. 该接口的权限是否正确，说下理由；
2. 请求的url格式 1）post/get ；2）entity格式；3）action格式；
3. 请求body字段的结构，包装过/平铺；
4. 错误码是包含了所有的了吗？
5. sdk注释了吗？