



物联网数据分析 最佳实践

文档版本: 20220329



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。 如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。	⚠ 危险 重置操作将丢失用户配置数据。
○ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔) 注意	用于警示信息、补充说明等 <i>,</i> 是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等,不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]
{} 或者 {a b}	表示必选项,至多选择一个。	switch {active stand}

目录

1.快速搭建室外停车场三维模型	05
2.快速接入设备位置到地图	10
3.在二维可视化页面嵌入第三方URL	17
4.分析透视之SQL分析决策直达设备	27

1.快速搭建室外停车场三维模型

本文描述如何使用物联网数据分析中的三维数据可视化功能,快速搭建室外停车场。

前提条件

在您的电脑设备上,下载并安装模型编辑器。

一、搭建三维模型

根据搭建工具手册,在模型编辑器中创建一个模型。
 使用栅栏墙圈地,并找到交通模块中的马路、停车位,放置好合适的位置。



2. 为模型的停车位增加 ObjectType=IOT 的属性,方便被物联网平台识别为一个设备与地磁感应器关 联。



车位较多时,可以复制模型使用。

C CampusBuilder(Momoda) v1.11.1608				
文件 编辑 选择 工具 视角 播件 帮助				
		i		Q
< 返回 自建筑 20	Q 请输入	搜索内容		
	Ⅲ 交通设	施(带电)		生物
	ſ		Ţ	い 10 生活
	Τ	0	Ţ	交通
	Î	Î	Ĩ	》 消皮
			ø	机房
				 其它
	l.	y	Ĭ	倒線
		En.		设备
	Ⅲ 交通设	施 (不带电)	1
	Ⅲ 交通工	具		个人
	Ⅲ 禁令标	志		☆ 功能
	111 弦生振	+		

3. 场景搭建完成之后保存并导出ThingJS场景包。



二、创建产品和设备

1. 参考创建产品内容, 创建地磁感应器产品。

← 创建产品 (设备模型)	
* 产品名称	
地磁感应器	
* 所属品类 💿	
智能城市 / 公共服务 / 地磁检测器	查看功能
* 节点类型	
连网与数据	
* 连网方式	
WiFi	,
* 数据格式 💿	
ICA 标准数据格式 (Alink JSON) ~	,
✔认证方式	

2. 参考批量创建设备内容,为刚刚创建的地磁感应器添加设备。

批量添加设备		×
* 产品		
地磁感应器		\sim
* 添加方式 自动生成 / 批量上传		
* 设备数量 💿		
10		
	确认	取消

3. 参考设备分组内容, 创建一个分组, 并将地磁感应器的所有设备添加到该分组中。

← 停车场						
分组层级 分组 / 停车场		分组ID	Sector A	复制		
设备总数 10		激活设备	0			
当前在线 0						
分组信息 设备列表 子分组列表						
満加公常到分组 全部产品	Q					C
DeviceName	设备所属产品		节点类型	状态/启用状态 🖓	最后上线时间	操作
385fkQ7m1QFA1HaP8BZR	地磁感应器		设备	● 未激活	-	查看 人分组中删除
mxhRUHwPNgCqtlCumB0s	地磁感应器		设备	◎ 未激活	-	查看 从分组中删除
uNXEz0hJ8wTrk4DMaqTf	地磁感应器		设备	 未激活 	-	查看 从分组中删除
uikor4msSNUcVxBm6x9H	地磁感应器		设备	◎ 未激活	-	查看 从分组中删除
11dAIFhVmHDjXCcfgaO5	地磁感应器		设备	◎ 未激活		查看 从分组中删除
AhQ0VQXhKCls5JdlQvx7	地磁感应器		设备	◎ 未激活	-	查看 从分组中删除
DmlpS662j81wellalqmk	地磁感应器		设备	◎ 未激活	-	查看 从分组中删除

4. 下载设备端SDK, 激活已创建的地磁感应器产品下的所有设备。

三、创建并分享场景

 所有设备激活成功后,创建三维数据可视化场景,关联上一步中创建的设备组,导入已制作完成的 ThingJS场景包。操作详情及参数说明请参见三维数据可视化。

创建场景	\times
 如何关联设备模型?点击查看文档 	
* 场景名称: 停车场	
* 目标广品以设备组:	
* 授权形式: 只读(展示) 读写(远程控制) * 场暑横型	
云端创建(推荐) 存年场模型.tjs 本地创建 ×	
确定取	消

2. 在三维数据可视化场景中,模型中的设备关联已在物联网平台创建的设备。

空间数	据可视化 > 三维数据可	视化			
停车	场				新手引导编辑 预览 分享
目标设	备组:停车场		授权形式:只读(展示)	设备总数: 10	模型来自:本地
Ð	型 关联设备				⑦ 场景模型 □ 事件告警 @ JS开发
	关联列表				
	✓ Campus				and the second se
	停车位	Ø	UJgfw2iz8M803Js \vee		
	停车位	Ø	rov2LPIpEk95pA0v >		A CONTRACTOR OF THE OWNER OWNER OF THE OWNER OWNE OWNER OWNE
	停车位	Ø	选择关联设备 ^		
	停车位	Ø	 > 路灯 > 地磁 		
	停车位	Ø	 UJgfw2iz8M80: 		
	停车位	Ø	 rov2LPIpEk95p. 7JsnsowaBi2m 		
	停车位	Ø	选择关联设备 ~		
					□ 标准视图

3. (可选)在三维可视化场景右上角单击**JS开发**,增加如下message监听代码,获取三维数据可视化场景 发送的设备属性数据,方便您做更加定制化的功能开发。

```
window.addEventListener('message', e => {
  const { data: { action, payload } } = e;
  if (action === 'deviceStatus') {
     console.log(payload);
  }
});
```

根据创建场景时设置的刷新频率, message监听代码会收到场景中所有设备的数据,并打印数据到浏览器控制台(Console)中,如下图所示。您可以根据数据处理您的3D模型。

Elements Console	Sources Network Performance	Memory Application Security Audits	🔹 📥 💷 🗄 🗙
▶ ⊘ top ▼	• Filter	Default levels 🔻	8 hidden 🏼 🏟
 Hool The "spa-tanputer" Hool The "spa-tanputer" (Invicotitate(List) Arms (Invicotitate(List) Arms Hool The "spa-tanputer" 	" will be deleted uses, plasse up " will be deleted uses, plasse up #200 " will be deleted uses, plasse up	e 'Agg, Fool, Camputes' to replace 21 e 'Agg, Fool, Camputes' to replace 21	Directored and a construction of the construct
i consular lagditty			
<pre> {deviceStatusList: Arra, vdeviceStatusList: Arra b0: {iotId: "SRy0 v1: iotId: "iXJw bstatus: {water_lew }_proto_: Object length: 2 }_proto_: Object </pre>	y(2)} ay(2) 000101", status: {…}} 000101" el: 5}		bl.js:1

参考空间数据可视化分享场景,分享已配置好的三维数据可视化场景。通过分享的场景观察停车场现状,智能化管理停车场业务。

2.快速接入设备位置到地图

本文描述如何使用物联网数据分析中的二维数据可视化功能,快速接入设备位置到地图。一般在空气质量检测、基站分布、资产管理等场景,需要接入设备位置到地图上,实时展示设备的运行状态,查看并管理设备。

步骤一、创建产品

- 1. 登录物联网平台控制台。
- 2. 参考创建产品, 创建带有位置属性的产品。

← 创建产品 (设备模型)				
* 产品名称				
test_product				
* 所属品类 🕜				
○ 标准品类 ● 自定义品类				
* 节点类型				
直连设备				
连网与数据 * _{连网方式}				
WiFi				
* 数据格式 📀				
ICA 标准数据格式 (Alink JSON)	-			
✔认证方式				
更多信息				
▼产品描述				

3. 参考单个创建设备,添加设备。

添加设备 💿	×
1 特别说明: deviceName可以为空,当为空时,阿里云会颁发全局唯一标 识符作为deviceName。	
产品 test_product	*
DeviceName 💿	
备注名称 ②	
请输入备注名称 确认 取消	

保存设备的设备证书(ProductKey、DeviceName、DeviceSecret),以备后续使用。

4. 参考<mark>设备属性上报位置(Geolocation)</mark>内容,为设备所属产品添加地理位置功能,并上报设备位置到云端。

添加标准功能			×
选择功能:	全选添加	已选功能:	全选删除
自定义品类	的类型		
地理位置	Q 🕲		
地理位置(属性) 标识符:GeoLocation 适用类别:CuttingN	/achine		
地理位置 <u>属性</u> 标识符: GeoLocation 适用类别: ArcExting	suishing		
地理位置(属性) 标识符:GeoLocation 适用类别:Agricultu	ıralMonitor		
地理位置(属性) 标识符:GeoLocation 适用类别:Conversi	onGateway		
	•		
			确定 取消

步骤二、模拟设备

使用Link Kit SDK提供的Java SDK Demo模拟设备的GPS模块。

- 1. 下载Java SDK Demo并在本地解压。
- 2. 在 device_id.json文件中,填写已保存的设备证书(ProductKey、DeviceName、DeviceSecret)信息。

🔲 Project 👻 😤 荣 —	💿 MqttSample.java 🗴 💿 ThingSample.java 🗴 🚭 HelloWorld.java 🗙 😨 DeviceInfoData.java 🗙 😨 LabelSample.java x 👔 🏭 device.jd.json 🗴
 ▼ Marchaeline Autoparticity > Marchaeline Autoparticity 	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 ▼ iava ▼ Im com.aliyun.alink.devicesdk.demo BaseSample COTASample DeviceInfobata DeviceShadowSample FileVitis GatewaySample HelleWorld LabelSample MattSample SubThingSample ThingSample ThingSample 	6 Sublevice': { 8 9 9 9 9 9 9 9 9 9 9 9 9 9
JavaLinkKitDemo.iml	

3. 在test_case.json文件中,设置设备要上报的位置属性。

由于在物联网平台控制台,位置属性产品的**产品详情 > 功能定义**页面中设置的GeoLocation属性定义 是struct结构体,因此*test_case.json*文件中的value参数需要填结构体子属性的JSON串。可单击子属性 名字后的查看获取其标识符。

查看标准功能		×
* 功能类型		
属性		
* 功能名称 🕘		
地理位置		
* 标识符 ②		
GeoLocation		
* 数据类型		
struct (结构体)		
* ISON羽ナキ・		
参数名称: 经度		查看
参数名称:纬度		查看
参数名称: 海拔		查看
参数名称: 坐标系统		查看
* 读写类型		
◎ 读写 ◎ 只读		
描述		
请输入描述		
		0/100
	确认	取消

test_case.json文件中,填好结构体子属性的JSON后,文件内容如下所示。



- ⑦ 说明 可使用高德开放平台经纬度拾取工具,设置设备经纬度(Longitude、Latitude)。
- 4. Demo程序的入口在*HelloWorld.java*文件中,本Demo中我们只需要设备上报位置的属性,因此需简化*H elloWorld.java*代码。简化后的代码如下所示。

```
public class HelloWorld {
    private static final String TAG = "HelloWorld";
```

```
private String pk, dn;
   private ThingSample thingTestManager = null;
   public static void main(String[] args) {
       ALog.d(TAG, "Hello world!");
       ALog.setLevel (ALog.LEVEL DEBUG);
       HelloWorld manager = new HelloWorld();
       ALog.d(TAG, "args=" + Arrays.toString(args));
       System.out.println(System.getProperty("user.dir"));
       String diPath = System.getProperty("user.dir") + "/device id.json";
       String deviceInfo = FileUtils.readFile(diPath);
       if (deviceInfo == null) {
          ALog.e(TAG, "main - need device info path.");
           return:
       }
       Gson mGson = new Gson();
       DeviceInfoData deviceInfoData = mGson.fromJson(deviceInfo, DeviceInfoData.class
);
       if (deviceInfoData == null) {
           ALog.e(TAG, "main - deviceInfo format error.");
           return;
       }
       ALog.d(TAG, "测试一机一密和物模型");
       manager.init(deviceInfoData);
    }
   public void init(final DeviceInfoData deviceInfoData) {
       this.pk = deviceInfoData.productKey;
       this.dn = deviceInfoData.deviceName;
       LinkKitInitParams params = new LinkKitInitParams();
       /**
        * 设置Mqtt初始化参数
        */
       IoTMqttClientConfig config = new IoTMqttClientConfig();
       config.productKey = deviceInfoData.productKey;
       config.deviceName = deviceInfoData.deviceName;
       config.deviceSecret = deviceInfoData.deviceSecret;
       config.channelHost = pk + ".iot-as-mqtt." + deviceInfoData.region + ".aliyuncs.
com:1883";
       /**
        * 是否接受离线消息
        * 对应mqtt的cleanSession字段
        */
       config.receiveOfflineMsg = false;
       params.mqttClientConfig = config;
       /**
        * 设置初始化设备证书信息,用户传入
        */
       DeviceInfo deviceInfo = new DeviceInfo();
       deviceInfo.productKey = pk;
       deviceInfo.deviceName = dn;
       deviceInfo.deviceSecret = deviceInfoData.deviceSecret;
       params.deviceInfo = deviceInfo;
       /**
        * 设置设备当前的初始状态值,属性需要和云端创建的物模型属性一致。
        * 如果这里什么属性都不填,物模型就没有当前设备相关属性的初始值。
```

```
* 用户调用物模型上报接口之后,物模型会有相关数据缓存。
        */
       Map<String, ValueWrapper> propertyValues = new HashMap<String, ValueWrapper>();
       // 示例
       // propertyValues.put("LightSwitch", new ValueWrapper.BooleanValueWrapper(0));
       params.propertyValues = propertyValues;
       thingTestManager = new ThingSample(pk, dn);
       LinkKit.getInstance().init(params, new ILinkKitConnectListener() {
           public void onError(AError aError) {
               ALog.e(TAG, "Init Error error=" + aError);
            }
           public void onInitDone(InitResult initResult) {
               ALoq.i(TAG, "onInitDone result=" + initResult);
               thingTestManager.readData(System.getProperty("user.dir") + "/test case.
json");
               thingTestManager.report();
           }
       });
   }
}
```

5. 执行HelloWorld.java文件,使设备的位置生效。

步骤三、创建二维数据可视化场景

1. 参考物联网平台设备展示, 创建一个数据来源为物联网平台的场景。

空间数据可视化			
二维数据可视化 三维数据可视化			
仅支持定义了物模型的产品使用此功能,建议配合	创建场景	×	
+	 如何配置设备位置信息?点击直着文档 场层名称: 		STOR
创建场展	testt汤最 # 目標产品或设备组:	1.5	THE REAL PARTY OF THE REAL PARTY OF THE REAL PARTY OF THE
	▶ Weiskel test_product ✓	<u> </u>	O 2019-07-10 17.07.18
	* 授权形式: 快读(展示) 读写(远程控制) 		
© 2019-05-08 19:14:03			
			〈上一页 1 下一页〉

其中,目标产品或设备组选择产品,并从下拉框中选择已创建的位置属性产品。

2. 进入二维可视化场景页面查看设备实时位置。



至此,您已经将您的物联网设备快速的接入到地图中了。

后续步骤

可根据分享场景内容,将已创建二维数据可视化场景分享给其他人。

3.在二维可视化页面嵌入第三方URL

本文描述如何在物联网数据分析中的二维数据可视化场景中,嵌入第三方连接。例如,二维场景中有监控摄 像头等设备,且设备本身具有自定义气泡弹框需求等场景,可以在二维场景中嵌入URL,在查看设备的运行 状态、管理设备的同时查看监控画面。

实现此功能,需要二维数据可视化场景与IoT Studio中的Web可视化配合使用,详细步骤请见下文。

步骤一: 创建产品并添加设备

- 1. 创建一个产品。本文以test_product为例。详细指导请参见创建产品。
- 2. 在产品详情页面,为test_product产品添加自定义功能。详细指导请参见单个添加物模型。

物联网平台 / 设备管理 / 产品 / 产品详情 / 功能定义	
 ・	 添加自定义功能 ・功能类型 ● ■ ■ ・功能名称 ● 三方连接
	 标识符 ● WEB_URL 数据类型 text (字符串) >数据长度: 2048 字节
	 ・读写英型 ④ 读写 ○ 只读 描述 请输入描述 0/100 入前込 取消

3. 为test_product产品添加标签。详细指导请参见标签。

物联网平台 / 认	设备管理 / 产品	/ 产品详情							
← test_	produc	t							
ProductKey		复制						ProductSecret	******* 查看
设备数	0 前往管理								
产品信息	Topic类列表	功能定义	数据解析	服务端订阅	设备开发				
产品信息	∠ 编辑								
产品名称	test_produc	:t				节点类型	直连设备		
所属品类	自定义品类				编辑	示签			×
动态注册 📀					产品标	<u>签</u>			
产品描述					stuc	dioType	url		删除
r					+ 新墙	标金			
标签信息 产品标签:无标	∠ 编辑 <							确认	2消

其中,标签key设置为*st udioType*,标签value有如下两种设置方法,不同的value,会影响第三方弹窗的弹出方式:

○ 标签value设置为url: 单击设备气泡弹出气泡窗口, 再通过气泡窗口中的URL链接弹出第三方弹窗。

○ 标签value设置为pop: 直接单击设备气泡弹出第三方窗口。

两种弹出方式的区别请见步骤三中的第8小步。

4. 为产品添加设备。详细指导请参见单个创建设备。

添加设备 💿	\times
()特别说明: deviceName可以为空,当为空时, 阿里云会颁发全局唯一标识符作为deviceName。	
产品	
test_product	
DeviceName 💿	
test02	
备注名称 ◎ 请输入备注名称	
确认 取消	肖

步骤二、创建二维数据可视化场景

1. 为设备设置地理位置。详细指导请参见设备属性上报位置(Geolocation)。

圣功能:	全选添加	已选功能:	全选删除
其他设备 其他	类型	地理位置 属性 标识符: GeoLocation	×
地埋位董	Q 🛛		
地理位置(<u>属性</u>) イ 标识符: GeoLocation 适用类别: CuttingN	lachine		
地理位置(<u>属性</u>) 标识符:GeoLocation 适用类别:ArcExting	Juishing	> <	
地理位置(属性) 标识符:GeoLocation 适用类别:Agricultu	ralMonitor		
地理位置(属性) 标识符:GeoLocation 适用类别:Conversio	onGateway		
	-		

2. 使用test_product产品创建一个二维数据可视化场景。详细指导请参见物联网平台设备展示。

创建场景	\times
• 如何配置设备位置信息? 点击查看文档	
 * 场景名称: LD测试 * 目标产品或设备组: 产品 设备组 test_product * 授权形式: ④ 只读(展示) (远程控制) 	
确定 上一步	

3. 设置设备的WEB_URL属性的值为设备需要链接到的第三方页面地址。可以使用**在线调试**功能,上报属性 值。

物联网平台		物联网平台 / 监控运维 / 在线调试	
概览		在线调试	
设备管理	^	请连择设备: test_product	
产品		调试真实设备 调试虚拟设备	实时日志 ● 在线 (虚拟设备
设备		属性上报 事件上报 属性调试 服务调用	类型/时间
分组		WEB_URL	
CA证书		https:/	
规则引擎	\sim		
监控运维	^		
实时监控			
运维大盘			
在线调试			
日志服务			
固件升级			
远程配置			
告警中心 New			
边缘计算	~		
视频服务	~		
IoT Studio	\sim		
数据分析 🖸			
应用托管	~		
曰 新版反馈		推送 策略推送 关闭虚拟设备 查看数据	

添加第三方链接后,单击二维数据场景中设备气泡后,如下所示。

空间数据可视化 > 二维数据可视化			
LD测试			批量导入位置 💿 編組 预览 分享
目标产品:test_product	授权形式:只读取设备状态	刷新频率:15秒	设备总数:2
▲ 第四次 ● ▲ 第四次 ● <p< td=""><td>床 潤 路 杭州微胜机械 物直有限公司</td><td>数 1 番 金会</td><td>西 中 古 市 市 市 市 市 市 市 市 市 市 市 市 市</td></p<>	床 潤 路 杭州微胜机械 物直有限公司	数 1 番 金会	西 中 古 市 市 市 市 市 市 市 市 市 市 市 市 市
	日花山村	Costa Coffee 25	L.

步骤三、IoT Studio可视化大屏搭建

IoT Studio可视化相关详细内容,请参见Web可视化开发。

1. 拖动二维数据可视化场景(地图组件)到IoT Studio大屏中,占满整个屏幕。



2. 添加一个if rame组件到大屏中,用于在点击设备气泡时,弹出一个第三方窗口。



3. 配置变量的联动关系。

为了点击设备气泡的时候,能够打开第三方地址,需要设置地图和if rame组件的交互关系,让if rame组件中设置的地址能够从设备传入if rame。



i. 单击地图,新增一个**交互**,并设置动作为*赋值给变量*。

ii. 单击管理变量 > 新增变量。

设置变量的名称,默认值设置为第三方URL。

新增变量		×
	变量名称: *	
	第三方链接	
	默认值:	
	https:/ =a2c	
	描述:	
	请输入变量说明,限100字	
	确认	取消

iii. 配置变量。



4. 单击iframe组件,单击配置,设置关联链接。



里击 参数米 源,选择 受 量,选择	匕配直灯旳吊二万斑按受重。
----------------------------------	---------------

配置链接		×	
因安全政策原因,目前 HTTP地址,需要绑定	前只支持HTTPS地址, 自己的域名才能够预览	如果您填写了 8	
链接内容		参数来源	
+ URL参数 🕜		变量 组件值	
选择变量		•	×
变量名	描述	默认值	操作
第三方链接	-	https:/	赋值来源 编辑
管理变量			确 定 取消

5. 因为只有点击设备气泡时,才显示第三方窗口,并且第三方窗口显示后,才可看到关闭按钮,因此需要 默认隐藏组件。关闭if rame组件和按钮组件的可见性。



6. 单击地图, 配置隐藏组件与地图的交互关系, 实现单击设备气泡时, 弹出三方页面和关闭按钮的功能。



7. 为按钮组件添加交互,实现单击关闭按钮后关闭iframe。



- 在界面右上方单击保存,然后单击预览,查看已配置完成的Web可视化大屏。
 根据产品标签value的不同设置,第三方窗口的弹出方式不同:
 - 标签value为*url*:单击设备气泡,在弹出窗口中单击第三方连接,则显示第三方弹窗,单击关闭按
 钮,可关闭弹窗。



○ 标签value为pop: 单击设备气泡,可直接弹出第三方窗口。



9. 切换到IoT Studio Web可视化开发编辑页面,在页面右上方单击发布,填写版本内容,单击确定。



系统显示发布成功后,处理Web可视化应用的后续操作请参见Web可视化开发。

4.分析透视之SQL分析决策直达设备

物联网数据分析提供的数据分析透视中SQL分析功能,不仅支持分析设备上报云端的数据,还可以通过SQL 将分析后的决策结果下发至设备进行远程控制。

前提条件

- 仅支持使用真实设备或使用SDK模拟设备实现下发决策至设备的功能。
- 下发数据的设备属性,必须是读写类型。

背景信息

本文以使用Java SDK模拟空调设备为例,利用物联网数据分析提供的数据开发功能编写SQL,从某一个提供数据的产品获取设备数据,并将该数据下发到空调设备。

操作步骤

- 1. 登录物联网平台控制台。
- 2. 创建一个空调设备。
 - i. 参考创建产品, 创建如下图所示的空调产品。

物联网平台 / 设备管理 / 产品 / 创建产品
← 创建产品 (设备模型)
* 产品名称
空调
* 所属品类 ②
● 标准品类 ○ 自定义品类
智能生活 / 大家电 / 空调机 💙 査看功能
* 节点类型
直连设备 か M关子设备 M关子设备
连网与数据
* 连网方式
WiFi 🗸
* 数据格式 ②
ICA 标准数据格式 (Alink JSON)
◇认证方式
更多信息
◇ 产品描述
保存 取消

产品创建成功后,到产品详情页面获取ProductSecret,以备后续使用。

ii. 参考单个创建设备,添加空调设备。

添加设备 💿	\times
(特别说明: deviceName可以为空,当为空时,阿里云会颁发全局唯一标 识符作为deviceName。	
产品	
空调	
DeviceName 💿	
airconditioner_1	
备注名称 ②	
请输入备注名称	
确认 取消	

设备创建成功后,保存设备证书(Product Key、DeviceName、DeviceSecret),以备后续使用。

- 3. 使用Link Kit SDK提供的Java SDK Demo模拟空调设备,使其上线。
 - i. 下载Java SDK Demo并解压到本地。
 - ii. 解压后在*device_id.json*文件中,填写已保存的设备信息(ProductKey、DeviceName、 DeviceSecret)和产品ProductSecret。



iii. 启动HelloWorld.java文件中的main方法。



返回物联网平台控制台,在设备管理页面选择空调产品后,可看到产品下的设备状态为在线。

物联网平台 / 设备管理 / 设备						
设备						
空调 イ	设备总数	• 激活设备 @ 1	● 当前在线 ◎ 1			C
设备列表 批次管理						
添加设备 批量添加 Device	Name ~ 遺輸入DeviceName	Q 请选择	平设留标签 💙			
DeviceName/衢注名称	设备所履产品	节点类型	状态/启用状态 ♀	最后上线时间	操作	
airconditioner_1	空调	设备	• 在线 🜑	2020/04/09 19:17:39	查看 删除	
動除 禁用 启用						

4. 创建商场温度传感器设备,并上报数据到云端。

i. 参考创建产品, 创建一个商场温度传感器产品。

物联网平台 / 设备管理 / 产品 / 创建产品	
← 创建产品 (设备模型)	
* 产品名称	
商场温度传感器	
* 所属品类 📀	
● 标准品类 ○ 自定义品类	
智能园区 / 室内温度传感器	查看功能
* 节点类型	
连网与数据	
* 连网方式	
WiFi	
* 数据格式 ②	
ICA 标准数据格式 (Alink JSON) く	
◇认证方式	
更多信息	
✓产品描述	
保存取消	

ii. 参考批量创建设备,批量添加5个商场温度传感器设备。

iii. 参考单个添加物模型,为商场温度传感器添加自定义功能。

? 说明 可根据您的需求添加多种属性,但需要下发的属性,其读写类型必须是读写。

iv. 在物联网平台,选择**监控运维 > 在线调试**,在**在线调试**页面,选择调试产品和设备后上报属性数据。

物联网平台		物研究平台 / 监控运输 / 在线调试		
概察		在线调试		
0.66.09.00		请远探设备: 网络温度传感器 >		
No III III A			実时日志 ● 在线(虚拟设备)	二日 (1997年) (199775) (199775) (199775) (1997755) (19977555) (1997755555) (199775555555555555555555555555555555555
/~~##		WEIT, MALAN, KETAR		
设备		羅性上报 事件上报 羅性漂流 服务调用	美型 / 时间	内容
分组		Areald	设备上报数据	2020-04-09 19:18:27.176, 48897274295148FA966C8688EE2DBC43, upstream - bizType=PR 0
CA证书		-12	2020/04/09 19:18:27	or "uniNg/d" 4922343950394624" ventil code 200 metagesuccess et response=["code":200_data:10_id":123", mesage:1:success "method":1thing.event.prope d
规则引擎	~	CurrentHumidity		evi gant 10 885 r. Alson Henrosou autoritus in granni an gran i, sastar vi sunta
监控运绳	~	调输入参数 (double) ◎		846 · · · · · · · · · · · · · · · · · · ·
实时监控		CurrentTemperature		
运输大盘		28		
在线调试				
日本服务				
E145465				
E014-7758X				
远程配置				
告替中心 New				
边缘计算	~			
视频服务	~			
IoT Studio	~			
应用开发				
而日繁華				
100 m 100 m		推送 策略推送 关闭虚拟设备 查看数据		

商场温度传感器产品下的5个设备,分别上报数据,可上报相同的属性值。

5. 在数据分析 > 数据资产页面,查看空调设备和商场温度传感器设备的数据表。

您可直接输入产品名称关键字,模糊搜索到相关数据表。在商场温度传感器和空调的数据概览页面中复制\$product_key的值,并查看设备属性标识符,以备后续SQL语句中使用。

数据资产 / 数据表										
← 数据表										
Q.南场	表信息	数据概范								
■ 南北高温度传感器	数据预览 默认预选编	说明 謝的20张数据。	如需查看更多数据,可以在可	視化分析或者SQL分析中重要。						
	\$device_na	ime	CurrentHumidity	\$event_time	CurrentTemperature	Areald	Sevent_date	\$product_key	\$iot_id	\$type
	10.000	0.00		1586432973199	28	-#	20200409	1000		DEVICE
	1000			1586432960606	28		20200409		Color of the local division of the local div	DEVICE
	1000			1586432947363	28		20200409	a property and	the second second	DEVICE
				1586432934450	28	-15	20200409	1000	Contraction of the	DEVICE
	-	-		1586431107176	28	-15	20200409	A CONTRACT OF A	-	DEVICE

- 6. 创建数据开发任务。
 - i. 参考SQL分析, 创建一个下发数据至设备的任务。

■ 下发数据至设备 SQL数据分析工作台									上次	泉存于: 2	020-04-09 19	645:46	0	0	 1	0	B		0	٥
数据表	物标签)															CSV下號	機振导	入运	行历史
数描源类型	设备数据源	✓ 1																		
Q. 搜索数据表																				
> 平台系统表																				
◇ 平台设备表																				
> 设备数据表																				
> 设备快服表																				
> 设备事件表																				
> 授权表																				

ii. 在任务中编写如下SQL。

```
-- 以下示例中的函数均可在函数列表中找到
-- 查询商场温度传感器上报温度且上报时间在五分钟内,且高温数量传感器数量大于5
CREATE VIEW HIGH TEMPERATURE AS
   SELECT count (CurrentTemperature) AS high temp num, AreaId ,AVG (CurrentTemperatu
re) AS current temper
   FROM ${pk.Your ProductKey1}
   WHERE CurrentTemperature IS NOT NULL AND DATE DIFF('MINUTE', FROM UNIXTIME($even
t time/1000),NOW()) < 5 AND CurrentTemperature > 26
      GROUP BY AreaId HAVING count (CurrentTemperature) > 5;
INSERT INTO ${pk.Your ProductKey2.Your DeviceName2}
SELECT
1 AS PowerSwitch.
-- 根据当前温度调整风速等级
CASE current temper
   WHEN 35 THEN 5
   WHEN 32 THEN 4
   WHEN 30 THEN 3
   WHEN 28 THEN 0
   ELSE 0 END AS WindSpeed,
-- 根据月份切换风力类型 自动 - 0; 制冷 - 1; 制热 - 2; 通风 - 3; 除湿 - 4;
CASE
   WHEN MONTH (NOW ()) > 6 AND MONTH (NOW ()) < 11 THEN 1
   WHEN MONTH (NOW () ) > 11 AND MONTH (NOW () ) < 4 THEN 2
   ELSE 0 END AS WorkMode
FROM HIGH TEMPERATURE WHERE AreaId='一楼';
```

其中:

- FROM 语句中, Your_ProductKey1替换为商场温度传感器的ProductKey。
- INSERT INTO 语句中, Your_Product Key2和 Your_DeviceName2替换为空调设备的 Product Key和DeviceName。

⑦ 说明 语法解释如下:

```
INSERT INTO {目标表} SELECT {属性标识符} FROM {数据表/视图} WHERE {过滤条件} [LIMIT (数据条数限制)]
```

编写完成后需进行语法校验。



iii. 保存并运行SQL任务。

运行		上次服存于:	2020-04-09 20:31:10	C	œ	4	0		• •		ł
1 1 2 3 4 5 5 6 7 8 9 9 10 11 12 13 14 15 15 16 6 7 7 8 9 9 10 11 12 2 3 24	REATE VIEW HIGH_TENSEANDE AS FEEDERATE VIEW HIGH_TENSEANDE AS FEEDERATE VIEW HIGH_TENSEANDE AS high_teng_num, Areald , AVG(Current Member Current Emperature 1s Not Nucl. AND DATE_DIFF("HUNDIT", FROU_UND) MARKE Current Emperature 1s Not Nucl. AND DATE_DIFF("HUNDIT", FROU_UND) MARKE Current Emperature 1s Not Nucl. AND DATE_DIFF("HUNDIT", FROU_UND) CONTENT (INTO \$(pk.alsCickewsK.alrConditioner_1) (ICC - (ICC - (I	<pre>2 JETTRE20 ITemperature) A5 current_temper TUPE(Sevent_time/1000),HCM()) < \$ AHD CurrentTemperature > 26</pre>						(5)下統	建成导入	运行1	速
	- (+9)										
Down	5. <u> 5.</u>										
- ower	head to be a second	MindSpeed	Mode	Mada							

7. 查看 Hello World. java 文件中设备接收到的下发属性。



至此,您已完成利用数据开发SQL将数据下发至设备的操作。