

ALIBABA CLOUD

阿里云

快速入门

文档版本：20201111

 阿里云

法律声明

阿里云提醒您在使用或阅读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。

1. 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。
2. 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。
3. 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，是用户必须了解的内容。	 注意 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。	 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 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.使用虚拟设备搭建茶园监控应用	05
1.1. 概述	05
1.2. 创建项目和产品	05
1.3. 开发虚拟设备	06
1.4. 分析温度时序数据	07
1.5. 使用云数据库创建数据表	08
1.6. 开发获取温度数据的服务	09
1.7. 开发可视化茶园监控大屏	15

1.使用虚拟设备搭建茶园监控应用

1.1. 概述

通过使用物联网应用开发（IoT Studio）平台的Web可视化开发和业务逻辑开发工具，开发一个茶园的环境监控大屏。应用用户可在监控大屏中，查看在指定时间内，设备上报的温度值折线图。

物联网 物联网平台 IoT IoT Studio Web应用开发 业务逻辑开发 环境监测 监控大屏

设计思路

- 先在业务逻辑开发工作台，创建三个服务，分别是：
 - 一个设备触发服务：用于获取温度探测器上报的温度数据，并存储到云数据库RDS版MySQL数据表中。
 - 一个定时触发服务：用于每个整点统计前一小时的最高温度数据，并存储到另一个云数据库RDS版MySQL数据表中。
 - 一个HTTP接口：用于从RDS数据库中，筛选出指定时间段内，每小时的最高温度数据。
- 使用Web可视化开发工作台，开发一个Web前端应用。用于调用HTTP接口，获取指定时间段内，每小时的最高温度数据，并展示在折线图中。

监控大屏如下图。



操作步骤

1. [创建项目和产品](#)
2. [开发虚拟设备](#)
3. [分析温度时序数据](#)
4. [使用云数据库创建数据表](#)
5. [开发获取温度数据的服务](#)
6. [开发可视化茶园监控大屏](#)

1.2. 创建项目和产品

在物联网应用开发（IoT Studio）上创建物联网开发项目，并为项目导入或创建温度监测器产品和设备。后续将在该项目下，创建业务逻辑应用和Web应用。

创建项目

1. 登录[物联网应用开发控制台](#)，在左侧导航栏单击IoT Studio > 项目管理。您也可登录[物联网平台控制台](#)后，在左侧导航栏选择相关产品 > IoT Studio，进入IoT Studio控制台。
2. 在项目管理页，单击新建项目。



3. 在新建项目页，鼠标移动至新建空白项目区域，单击创建空白项目。
4. 在新建空白项目对话框，配置项目名称（例如[茶园监控方案](#)）和描述信息。



5. 单击确认。

关联已有产品和设备

若您已创建了产品和设备，且已为产品定义了物模型，直接关联产品和设备即可。

1. 在项目管理页面的项目列表中，找到目标项目（*茶园监控方案*），单击项目卡片。
2. 在左侧导航栏单击产品。
3. 在产品页面，单击产品列表左上方的关联物联网平台产品。
4. 在关联物联网产品页面的产品列表中，勾选待关联的产品。本示例中关联温度检测器产品及设备。



5. 选中关联产品同时关联其下所有设备的复选框。
6. 单击确定。
您可在该项目的产品页面的产品列表中查看到关联的产品。

新建产品和设备

若您还未创建相应产品，需先创建产品，并为产品定义物模型。

1. 在项目管理页面的项目列表中，找到目标项目，单击项目卡片。
2. 在项目页的左侧导航栏，单击产品 > 创建产品。本示例为产品命名为 *温度监测器*，选择自定义品类，其他参数使用默认值。

更多参数的说明请参见 [参数说明表](#)。



3. 单击保存。
产品创建成功后，回到产品页面，可在产品列表中查看到已创建的新产品。
4. 在产品列表中，单击产品名称，进入产品详情页，为产品定义功能。

本示例中，需为温度监测器产品定义一个温度属性（temperature）。

- i. 单击功能定义 > 自定义功能 > 添加自定义功能。
- ii. 在添加自定义功能对话框中，选择功能类型为属性，添加功能属性。



- iii. 单击确认。在自定义功能的功能列表中，您可查到已添加的温度属性。



5. 单击设备 > 新增设备，批量注册设备。本示例中以新增一个设备为例。
 - i. 单击新增设备。
 - ii. 在新增设备对话框，选择产品和添加方式。本示例选择 *温度监测器* 产品和自动生成方式，新增 1 个设备。
 - iii. 单击提交。
 - iv. 设备创建成功后，单击下载激活凭证，下载本批次设备的设备证书。

有关设备的更多详细内容，请参见 [设备](#)。

后续步骤

[开发虚拟设备](#)

1.3. 开发虚拟设备

物联网平台设备的正常开发流程是：设备端开发完成，设备上报数据，云端接收数据，云端开始开发工作。该开发流程战线较长，耗时较久。而虚拟设备可以快速模拟真实设备产生业务数据，是最快体验IoT Studio开发能力的途径。本案例根据开发虚拟设备的数据，完成应用的开发调试。

操作步骤

1. 登录[物联网应用开发控制台](#)，在左侧导航栏单击IoT Studio > 项目管理。您也可登录[物联网平台控制台](#)后，在左侧导航栏选择相关产品 > IoT Studio，进入IoT Studio控制台。
2. 在普通项目列表中，找到目标项目，单击项目卡片。您也可单击全局资源项目，进入该项目详情页面。
3. 在项目页面，单击左侧导航栏的设备。
4. 在设备列表上方，选择产品温度检测器。



5. 找到新增的温度检测器设备，单击设备右侧操作栏的查看。
6. 在设备详情页选择在线调试 > 调试虚拟设备 > 启动虚拟设备 > 属性上报。



7. 使用虚拟设备上报模拟属性值到云端。本示例中使用random()函数生成随机值来模拟设备属性值，以连续推送策略上报到云端。



执行结果

在设定的时间推送数据后，可在页面右侧实时日志下查看操作日志。



数据推送成功后，可在运行状态页签下，查看设备上报的属性信息。



有关调试虚拟设备的更多详细内容，请参见[调试虚拟设备](#)。

1.4. 分析温度时序数据

数据分析服务是物联网应用开发（IoT Studio）的一个重要组件，通过数据分析组件，可简单查阅各数据中各事件维度的统计值，例如平均值，最大值，最小值等。支持通过图表选择不同的数据维度，探索各数据之间的关系。

操作步骤

1. 登录[数据分析控制台](#)，在顶部导航栏上选择分析透视，进入时序透视页签，如下图所示。





 说明 上图中时序透视的快照图，需要在时序透视工作台中保存后才会生成。

2. 单击新建时序透视，在新建时序透视弹框中配置参数，如下图所示。



3. 单击确认，进入时序透视工作台，如下图所示。

 **说明** 本文图片所示为默认主题（亮色）。物联网数据分析支持设置工作台主题样式功能，您可单击工作台右上角的设置按钮，修改主题为暗色主题。



4. 单击**添加产品**，在**添加产品**对话框中选择需要添加的产品并单击**确认**，如下图所示。



5. 选择**设备优先**，在左侧属性选择区域选择**温度**，右侧出现如下图所示的折线图。



从上图中可以发现每隔1小时左右，温度的平均变化情况。您也可以查看温度上报数据的其他统计值，了解更多温度变化趋势。

后续步骤

为了能实时监控茶园环境温度变化趋势，您可以创建Web可视化大屏，直观展示温度数据。本案例以展示每小时内最高温度为例，介绍如何开发茶园监控大屏。

1. [使用云数据库创建数据表](#)
2. [开发获取温度数据的服务](#)
3. [开发可视化茶园监控大屏](#)

1.5. 使用云数据库创建数据表

在云数据库RDS版控制台，创建一个MySQL类型的数据库实例，并创建两个数据表，分别用于存储设备上报的温度数据和服务应用处理后的温度数据。

前提条件

已完成温度数据分析。更多信息，请参见[分析温度时序数据](#)。

操作步骤

1. 登录[云数据库RDS版控制台](#)。
2. 在云数据库管理页，单击**创建实例**，创建一个MySQL类型的数据库实例。
3. 在数据库实例列表中，单击该实例对应的**管理操作**按钮。
4. 在左侧导航栏中，单击**账号管理**，创建数据库用户账号和密码。用于登录数据库。
5. 在左侧导航栏中，单击**数据库管理**，创建数据库。
6. 在左侧导航栏中，单击**数据安全性**，添加数据库IP白名单。添加白名单具体操作，请参见[设置白名单](#)。
7. 在左侧导航栏中，单击**基本信息**，查看该数据库的基本信息。



8. 在数据库页面上方，单击**登录数据库**，输入用户账号和密码，登录数据库。
9. 在DMS左侧导航栏，选择已创建的数据库，在数据库中创建两个数据库表，分别命名为 sensor_data（用于存储各设备上报的温度数据）和Maxtemperature（用于存储服务应用处理后的数据）。

- 有关数据管理DMS更多信息，请参见[数据管理 DMS](#)

- 有关云数据库RDS版更多信息，请参见[RDS MySQL数据库](#)。

后续步骤

开发获取温度数据的服务


1.6. 开发获取温度数据的服务

在业务逻辑工作台，开发三个业务服务，分别用于获取设备上报的温度数据，每整点统计前一小时的最高温度数据，和筛选指定时间段内每小时的最高温度数据。

前提条件

已完成以下操作：

- [创建项目和产品](#)
- [开发虚拟设备](#)
- [分析温度时序数据](#)
- [使用云数据库创建数据表](#)

 **说明** 开发设备有不同的方法，用户可根据实际场景需要，选择相应的设备开发方法。本案例以开发虚拟设备为例。

操作步骤

1. 登录[物联网应用开发控制台](#)，在左侧导航栏，单击IoT Studio > 应用开发。您也可登录[物联网平台控制台](#)后，在左侧导航栏选择相关产品 > IoT Studio，进入IoT Studio控制台。
2. 在开发工具模块，单击**业务逻辑**。



3. 在**业务逻辑开发**页面，单击空白模板区域。



4. 创建一个设备触发服务。该服务以温度监测器设备作为触发节点，接收温度监测器设备上报的温度数据，并存储到云数据库RDS版MySQL数据表中。
 - i. 输入业务逻辑服务的基本信息并单击**确认**。

参数	说明
服务名称	服务的唯一标识符，在项目下具有唯一性。例如： <i>存储温度数据</i> 仅支持中文汉字、英文字母、数字、下划线（_）、连接号（-）和英文圆括号（()），且必须以中文汉字、英文字母或数字开头，长度不超过30个字符（一个中文汉字算一个字符）。
所属项目	服务所属的项目。本案例选择已创建的项目： <i>茶园监控方案</i> 。
备注	描述服务的用途等信息。长度不超过100个字符（一个中文汉字算一个字符）。

- ii. 从服务编辑器左侧**节点**页签下，拖拽一个**设备触发**节点到画布上，并配置输入数据为温度监测器上报的属性。




- iii. 配置一个云数据库RDS节点，用于将温度监测器上报的温度数据存入数据表sensor_data。

本示例中，输入的参数为：

```
{
  "table": "sensor_data",
  "rows": [
    {
      "id": "{{payload.deviceContext.gmtCreate}}",
      "deviceName": "{{payload.deviceContext.deviceName}}",
      "dateTime": "{{payload.props.temperature.time}}",
      "temperature": "{{payload.props.temperature.value}}"
    }
  ]
}
```

- iv. 单击部署，部署服务。
- v. 单击启动，启动服务。

 **说明** 本案例是以开启虚拟设备上报数据，无需再次调试。详细内容请参见[开发虚拟设备](#)。

5. 创建一个定时触发服务。

该服务用于每整点统计前一小时的最高温度数据，并存储到云数据库RDS版MySQL数据表中。

- i. 切换到服务列表页面，单击服务列表右侧的添加图标

- ii. 在业务逻辑开发页面，创建并设置定时触发服务。
 - a. 打开新建业务服务对话框，详细操作请参见[步骤3](#)。
 - b. 设置定时触发服务的基本信息设置（例如服务名称为存储最高温度数据），详细操作请参见[步骤4](#)的第一个子步骤。
- iii. 配置一个定时触发节点，用于每整点触发一次服务。

- iv. 配置一个云数据库RDS节点，用于从存储设备上报的温度数据的数据表sensor_data中，查询数据。

本示例中，查询数据的输入参数为：

```
{
  "table": "sensor_data",
  "condition": {
    "where": {"deviceName": "bZxNh9qi6PyqYvFcDvqK"},
    "columns": [
      "id",
      "deviceName",
      "temperature",
      "dateTime"
    ],
    "orders": ["temperature"]
  }
}
```

- v. 配置一个Node.js节点，用于提取每小时内，设备上报的最高温度。

本示例中，输入的脚本为：

```
/**
 * @param {Object} payload 上一节点的输出
 * @param {Object} node 指定某个节点的输出
 * @param {Object} query 服务流第一个节点的输出
 * @param {Object} context { appKey, appSecret }
 */
module.exports = async function(payload, node, query, context) {
  console.log("payload: ", payload);

  let result = {};
  result.index = payload.length - 1;
  result.maxTemperature = payload[result.index].temperature;
  result.id = payload[result.index].id;
  result.deviceName = payload[result.index].deviceName;
  result.dateTime = payload[result.index].dateTime;
  return result;
}
```

vi. 再配置一个云数据库RDS节点，用于将node.js处理后的数据存入数据表Maxtemperature。

本示例中，输入的参数为：

```
{
  "table": "Maxtemperature",
  "rows": [{
    "id": "{{payload.id}}",
    "deviceName": "{{payload.deviceName}}",
    "dateTime": "{{payload.dateTime}}",
    "maxTemperature": "{{payload.maxTemperature}}"
  ]
}
```

- vii. 单击部署，部署服务。
 - viii. 单击启动，启动服务。
 - ix. 单击调试，输入模拟时间，调试服务。
6. 创建一个HTTP接口，用于实现筛选指定时间段的数据。
- i. 参照步骤5中的子步骤1和2，新增一个服务（例如获取温度数据）。
 - ii. 配置一个HTTP请求节点，设置筛选温度数据API的Action名称和请求参数。

本示例中，配置了以下三个请求参数。

参数	类型	是否必需	说明
DeviceName	String	否	调用接口时，传入设备名称，查询指定设备上报的温度数据；不传入设备名称，则返回全部设备的温度数据。
startTime	Int	是	调用接口时，需传入要查询数据的起始时间。
endTime	Int	是	调用接口时，需传入要查询数据的结束时间。

- iii. 配置一个云数据库RDS节点，用于从数据表Maxtemperature中获取每小时的最高温度数据。

本示例中，输入的参数如下。

```
{
  "table": "maxtemperature",
  "condition": {
    "where": {
      "deviceName": "{{payload.DeviceName}}"
    },
    "columns": [
      "maxTemperature",
      "dateTime"
    ],
    "orders": [
      "dateTime"
    ]
  }
}
```

- iv. 配置一个Node.js节点，用于筛选指定时间段内的温度数据。

示例脚本如下。

```
/**
 * @param {Object} payload 上一节点的输出
 * @param {Object} node 指定某个节点的输出
 * @param {Object} query 服务流第一个节点的输出
 * @param {Object} context { appKey, appSecret }
 */
module.exports = async function(payload, node, query, context) {

  console.log("payload: ", payload);

  let length = 0;
  for (let i = 0; i < payload.length; i++) {
    if(parseInt(payload[i].dateTime) >= query.startTime && parseInt(payload[i].dateTime) <= query.
endTime) {
      length = length + 1;
    }
  }

  console.log("length", length);

  var result = new Array(length);
  for(let i = 0; i < length ; i++){
    result[i] = new Array(2);
    result[i][0] = "";
    result[i][1] = 0;
  }

  let valid = 0;
  for(let j = 0 ;j < payload.length ; j++){
    if(parseInt(payload[j].dateTime) >= query.startTime && parseInt(payload[j].dateTime) <= query.
endTime) {
      result[valid][0] = payload[j].dateTime;
      result[valid][1] = payload[j].maxTemperature;
      valid = valid + 1;
    }
  }

  return result;
}
```

- v. 配置一个HTTP返回节点，输出Node.js节点处理后的数据。

- vi. 单击**部署**，部署服务。
- vii. 单击**调试**，输入请求参数，调试服务。

7. 三个服务均调试成功后，单击页面右上角的发布按钮，发布服务。

后续步骤

开发可视化茶园监控大屏

1.7. 开发可视化茶园监控大屏

您可开发一个Web可视化应用，用于展示和查询指定时间段中，设备上报的每小时内最高温度。

操作步骤

1. 登录[物联网应用开发控制台](#)，在左侧导航栏，单击IoT Studio > 应用开发。您也可登录[物联网平台控制台](#)后，在左侧导航栏选择相关产品 > IoT Studio，进入IoT Studio控制台。

2. 在开发工具模块，单击**Web可视化开发**。

3. 创建可视化Web应用。

- i. 在可视化Web应用开发页面，单击空白应用区域。

- ii. 在新建Web应用对话框中，设置应用基本信息，单击**确认**。

参数	描述
应用名称	设置应用名称（例如： 茶园监控大屏 ）。支持中文汉字、英文大小写字母、数字、下划线（_）、连接号（-）和英文圆括号（()）；必须以中文汉字、英文字母或数字开头；长度不超过30个字符（一个中文汉字算一个字符）。
所属项目	选择已创建项目： 茶园监控方案 。
描述	描述该应用。长度不超过100字符（一个中文汉字算一个字符）。

4. 配置应用页面和标题。

- i. 设置页面背景和分辨率，可自定义；配置一个矩形组件，作为其他组件的背景。更多详细配置请参见[页面](#)和[矩形](#)。

- ii. 配置一个文字组件，用于展示标题（例如：[茶园温度检测](#)）。详细配置请参见[文字](#)。

5. 配置选择设备的下拉框。

- i. 配置一个文字组件，作为下拉框的标题。

- ii. 添加一个下拉框组件，配置其样式。下拉框中，显示设备名称。详细配置请参见[下拉框](#)。

参数	说明
列表内容	选择为设备，表示下拉框中展示设备名称。
选择产品	选择设备所属的产品。本案例选择： 温度检测器 。下拉框中，展示该产品下的设备名称。
默认值	可选设置。指定选项或始终选择列表第一项为默认项。

- iii. 选择配置栏中的交互，单击**新增交互**。
 iv. 选择事件为**值改变**；动作为**赋值给变量**，单击**管理变量**。
 v. 单击**新增变量**，新增一个名称为DeviceName的变量。

- vi. 返回交互页面，单击**配置 > 赋值**，选择**value**，赋值给变量DeviceName。

- vii. 单击**确定**，完成交互动作配置。

6. 配置时间组件，用于设置待查询数据所属的起始时间。

- i. 配置一个文字组件，作为时间组件的标题。
 ii. 配置时间组件样式。时间单位选择为**秒**。详细配置请参见[时间](#)。

- iii. 配置交互动作。创建一个变量startTime，并配置通过值改变事件，触发交互动作，赋值给变量。详细操作指导请参见步骤：[配置选择设备的下拉框](#)。

7. 参照上一步骤，再配置一个时间组件，用于设置待查询数据所属的结束时间，并配置交互动作。交互动作：值改变事件，触发交互动作，赋值给变量endTime。

8. 配置一个折线图组件，用于展示温度数据。下文有关折线图组件的详细配置，请参见[折线图](#)。

- i. 配置一个文字组件，作为折线图组件的标题。
 ii. 添加折线图组件，调整组件在页面的位置。
 iii. 配置折线图组件的数据源为已创建的HTTP接口（例如：获取温度数据），请求参数值设置为前面创建的变量。接口详情请参见[开发获取温度数据的服务](#)。

? 说明 如果接口返回数据不满足组件的数据格式要求，需将其转换为组件需要的数据格式。转换方法请参见[数据过滤器](#)。

- iv. 为折线图配置展示数据，完成配置。您可根据实际需要配置折线图的展示样式。

- v. 为结束时间的组件，配置交互动作2：值改变事件，触发交互动作，刷新折线图组件。

9. 单击顶部栏中的**保存按钮**，保存应用配置。

执行结果

1. 单击顶部栏中的**预览按钮**，预览应用。
2. 单击**发布按钮**，发布应用。