

为SIoT写一个物联网设备管理插件

谢作如 浙江省温州中学

SIoT是一款简单易用的MQTT服务器软件,也是“虚谷物联”项目的核心软件之一。SIoT不用安装,也不用注册、设置,一键运行后,其他物联网设备(如掌控板)按照一定的规范接入即可正常使用。正因为这一点,SIoT得到创客教师们的热捧,成为中小学物联网教学的重要支撑软件。

但是,能不能用SIoT做一个真实的物联网项目,如智慧农场、校园气象站、门禁系统等?我们会发现,功能实现并不难,但管理设备不太方便。SIoT毕竟是通用的MQTT服务器,在线设备管理之类的功能,只能自行开发,这样难度就有点大。因而,我冒出一个大胆的想法:能不能为SIoT编写一个插件,让这个通用的MQTT服务器变成可以商用的物联网系统管理平台?

● 为SIoT编写网页插件的可行性分析

作为SIoT开发团队的成员,我对SIoT的原理还是比较清楚的。SIoT运行后,会启动两种网络服务:一是MQTT服务,作用是处理MQTT消息;二是Web服务,用来运

行Web管理页面,实现在线管理。具体工作流程如图1所示。

SIoT是一款绿色软件,主目录下文件夹“static”中存放的是SIoT的Web管理页面,如下页图2所示。需要强调的是,SIoT的Web管理页面和SIoT服务器之间是利用WebAPI的方式交互数据的。也就是说,和我们传统的动态网站开发方式不同,Web管理页面并不是去读数据库,而是发送特定的Web请求,SIoT再返回相应的数据,实现页面管理功能。这种软件设计能够将不同的网络服务隔离,开发效率更高。

用记事本打开“static”中的HTML文件,如“devices.html”,

在源代码中可以看到其使用了目前最热门的Web前端框架vue.js技术,如下页图3所示。前端页面设计和后端数据隔离,是现在最主流的Web开发模式。那么,思路就呼之欲出了:只要在SIoT的管理页面上,按照既定的API规范,就能实现我们预设的功能了。如果同样采用Vue.js编写,那和现有的Web页面结合更加紧密。对于曾经开发过多个网站的资深网管来说,编写这样的页面,技术上不算太难。

● SIoT的API功能简介

我曾经在SIoT的文档中,提供过一个有趣的范例,即在PPT和Doc中插入超链接,就能通过SIoT控制掌控板的开灯、关灯,甚至可以

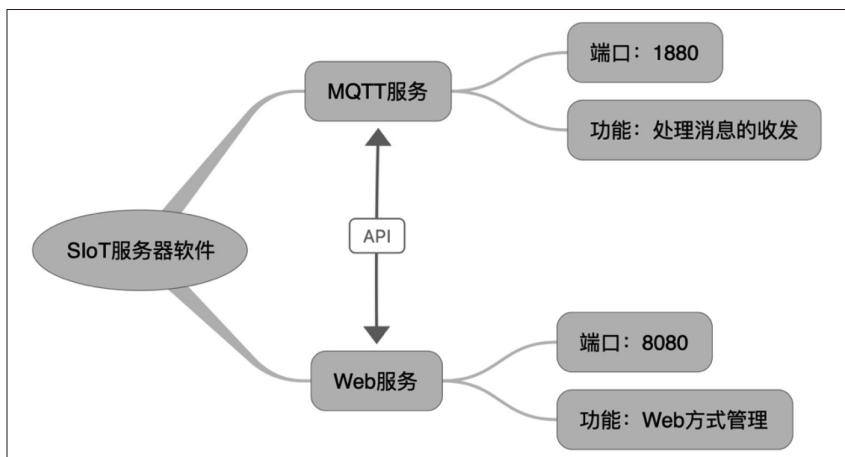


图1 SIoT的工作流程

和Mind+中的角色互动。这一功能的实现,其实就是利用了SIoT提供的WebAPI。只要通过浏览器访问特定的地址即可。

地址: `http://SIoT的IP:8080/publish?topic=xzr/001&msg=on&iname=siot&ipwd=dfrobot`。

这里的“SIoT的IP”指运行SIoT的电脑IP地址, `iname`和`ipwd`要替换为SIoT的用户名和密码。

那么, SIoT究竟提供了哪些API? 打开SIoT的使用文档, 在“高级操作技巧”的“3.WebAPI”部分, 可以看到具体的WebAPI清单, 如右表所示。

SIoT文档地址: `https://siot.readthedocs.io/`

● “智慧农场”管理插件的编写

我准备利用SIoT来开发一个学校智慧农场的项目。这个项目事先定义了物联网协议MQTT的各种细节, 如每个物联网终端节点的名称为“sf”加局域网IP地址的尾号, 终端上的传感器或执行器模块, 用连接的引脚编号来命名, 如a0、d2。这样一来, 每一个模块都有唯一的TopicID, 如“sf88/a0”表示名称为“sf88”的终端上的设备“a0”。

其中, “a0”接光线传感器, “a1”接土壤湿度传感器, “d2”接电磁阀。这个插件的作用是, 将所有的终端信息显示在一个页面上, 并能够用网页按钮的方式进行浇水,

SIoT的WebAPI一览表

类别	格式	说明
发送消息	<code>/publish?topic=xzr/001&msg=on&iname=siot&ipwd=dfrobot</code>	向topicid (主题) “xzr/001” 发送内容为“on”的消息, 其中“xzr”是项目id, “001”是设备id
获取最新数据	<code>/lastmessage?topic=xzr/001&iname=siot&ipwd=dfrobot</code>	获取topicid (主题) “xzr/001” 的最新一条消息
获取消息列表	<code>/messages?topic=xzr/001&iname=siot&ipwd=dfrobot&pn=1&psize=10&begin=2019-04-04&end=2019-07-01</code>	获取topicid (主题) “xzr/001” 从“2019-04-04”到“2019-07-01”的数据。其中时间可以省略
删除消息	<code>/clearmsg?topic=xzr/001&iname=siot&ipwd=dfrobot</code>	删除topicid (主题) “xzr/001” 的所有消息
获取项目列表	<code>/projects?iname=siot&ipwd=dfrobot</code>	一个项目中包含多个设备
更新项目	<code>/updateprj?pid=xzr&iname=siot&ipwd=dfrobot&desc=科学测量</code>	将名称为“xzr”的项目的备注修改为“科学测量”
获取设备列表	<code>/devices?pid=xzr&iname=siot&ipwd=dfrobot</code>	返回项目名称为“xzr”的设备列表
更新设备	<code>/updatedev?pid=xzr&dname=001&iname=siot&ipwd=dfrobot&desc=台灯控制</code>	将项目名称为“xzr”、设备名称为“001”的设备, 备注修改为“台灯控制”
删除设备	<code>/deldev?topic=xzr/001&iname=siot&ipwd=dfrobot</code>	删除名称为“xzr”的项目

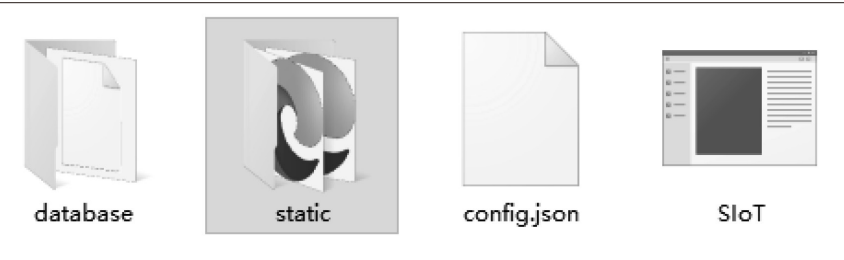


图2 SIoT的文件目录

```
devices.html — 已编辑

<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0" />
  <!-- 最新版本的 Bootstrap 核心 CSS 文件 -->
  <link rel="stylesheet" href="js/bootstrap/css/bootstrap.min.css" crossorigin="anonymous">
  <link rel="stylesheet" href="css/style.css" crossorigin="anonymous">
  <script src="js/jquery-2.1.4.min.js" crossorigin="anonymous"></script>
  <script src="js/jquery.cookie.min.js" crossorigin="anonymous"></script>

  <!-- 最新的 Bootstrap 核心 JavaScript 文件 -->
  <script src="js/bootstrap/js/bootstrap.min.js" crossorigin="anonymous"></script>
  <script src="js/vue.js"></script>
  <script src="js/app.js"></script>
</head>
<title></title>
<meta name="description" />
```

图3 devices.html文件的代码

也就是给“sf88/d2”的TopicID发送特定的消息。

第一步：新建网页

复制“devices.html”文件，重命名为“sfarm.html”。然后在“modules”文件夹中，也将“devices.js”文件复制一份，改名为“sfarm.js”。当然，每一个页面上都要做好超级链接，这些都是最基础的工作，不再赘述。

接下来，需要修改“sfarm.html”文件。核心工作是修改

“<section>”和“</section>”之间的内容。当然，这要求懂一点HTML语法和Vue.js。我也是边读Vue.js教程，边请教身边的专家，然后照猫画虎写的，如图4所示。

第二步：编写js代码

设计好网页后，最难的工作肯定是编写js代码。因为需要写的内容有点多，这里重点介绍“sendMsg_af”的函数。这是供“sfarm.html”上的按钮来使用的。如何实现“浇水”？其实就是发

送一个内容为“1”的消息(如图5)。

第三步：效果测试

运行SIoT，打开Web管理端，即可看到多了一个名为“智慧农场”的栏目。点击后，就会看到所有的设备都列出来了，在一个表格中显示了最新的光照和土壤湿度数据。点击“浇水”按钮，就能给相应的TopicID发送消息。

● 插件编写给“虚谷物联”带来了新玩法

这个“智慧农场”插件一经展示，就赢得了很多教师的好评和关注，有希望提供开发教程的，也有提出新的需求的。毫无疑问，这种开发插件的形式，给“虚谷物联”带来了新拓展思路。例如：编写一个校园气象站的插件，在一个页面上呈现最新的温度、湿度、光照、风向、风速等信息；编写一个会议室门禁管理插件，可以远程开门，可以遥控空调，还可以显示摄像头画面，等等。

我始终认为，只有在真实的情境中应用，创客作品的制作、STEM项目的研究才显得更有价值。给SIoT编写插件的最重要意义在于，将技术门槛降低了，物联网技术的应用显得越来越简单，开发真实应用不再困难。目前，SIoT的开发人员已经准备在下一版优化代码，让编写插件的技术难度变得更低。到那个时候，相信会有更多的中小学智慧校园系统中，运行着中小學生自主编写的各种智能应用。e

```
<section>
  <table class="table table-striped">
    <thead>
      <tr>
        <th>项目ID</th>
        <th>当前光照数据 (a0) </th>
        <th>当前土壤数据 (a1) </th>
        <th>操作</th>
        <th>项目说明</th>
      </tr>
    </thead>
    <tbody>
      <tr v-for="(project,k) in projectsList">
        <th scope="row">{{project.ID}}</th>
        <td :class="project.ID+'_a0'">{{getLastMsg(project.ID,"a0")}}</td>
        <td :class="project.ID+'_a1'">{{getLastMsg(project.ID,"a1")}}</td>
        <td>
          <button type="button" class="btn btn-default"
            @click="openurl(project.ID)">详情</button>
          <button type="button" class="btn btn-default"
            @click="sendMsg_af(project.ID + '/d2','1')>浇水</button>
        </td>
        <td>{{project.Description}}</td>
      </tr>
    </tbody>
  </table>
</section>
```

图4 “sfarm.html”中修改后的代码

```
sendMsg_af: function (af_topic, af_msg) {
  var that = this;
  if (this.loading == 1) {
    return;
  }
  this.loading = 1;
  app.get("/publish", {
    topic: af_topic,
    msg: af_msg,
  }, function (res) {
    that.loading = 0;
    if (res.code != 1) {
      app.alert(res.msg);
    } else {
      app.alert("指令发送成功");
    }
  })
},
```

图5 代码