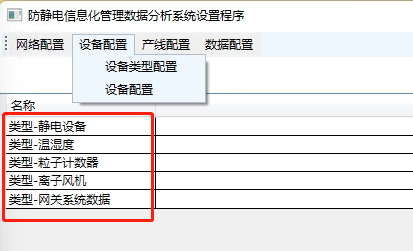
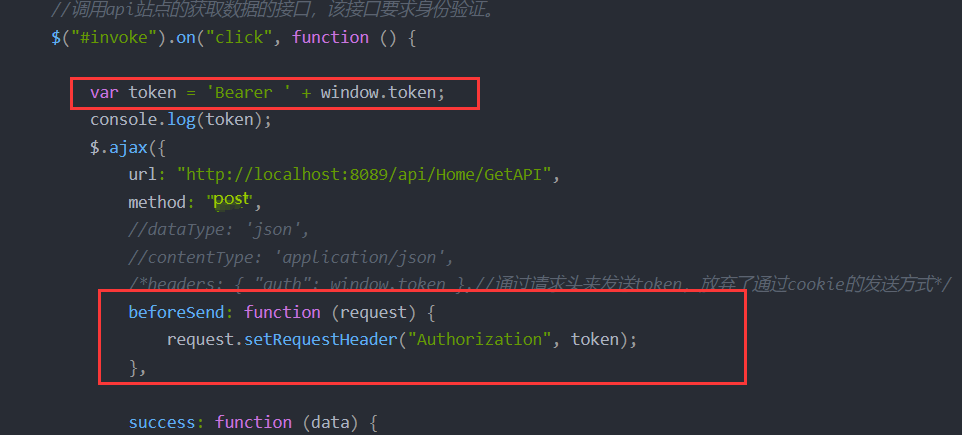
## 增加读写设备类型配置接口



1、前端历史曲线筛选、事件筛选需要用到。

2、后期扩展，前端可以修改设备类型。

数据包含： 类型唯一编号、类型名称



token由登录成功后签发

（图一）

设备类型查询接口：url/SystemSet/GetEquipTypeList

请求格式：post、header传递token(传递示例参考图一)

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{“EquipTypeId”:” 类型唯一编号”,”EquipTypeName”:”类型名称”}}

设备类型新增接口：url/SystemSet/AddEquipType

请求格式：post、header传递token(传递示例参考图一)

参数：{"EquipTypeName":"类型名称"}

返回格式：{“Status”:”状态”,” Message”:”消息” }

设备类型修改接口：url/SystemSet/ModEquipType

请求格式：post、header传递token(传递示例参考图一)

参数：{"EquipTypeId":5,"EquipTypeName":"类型名称1"}

返回格式：{“Status”:”状态”,” Message”:”消息” }

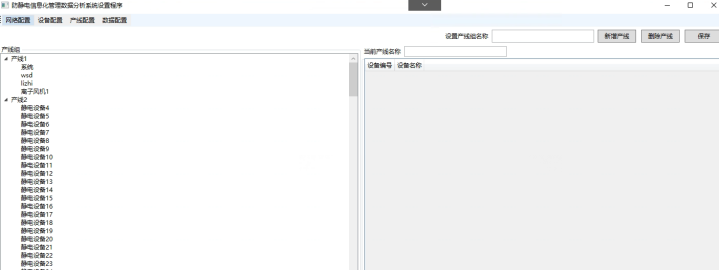
设备类型删除接口：url/SystemSet/DelEquipType

请求格式：post、header传递token(传递示例参考图一)

{"EquipTypeId":5,"EquipTypeName":"类型名称1"}

返回格式：{“Status”:”状态”,” Message”:”消息” }

## 增加读写产线分组配置接口



1、前端产线界面展示需要清楚有那些几个产线分组。每个产线分组下有那些数据需要展示。

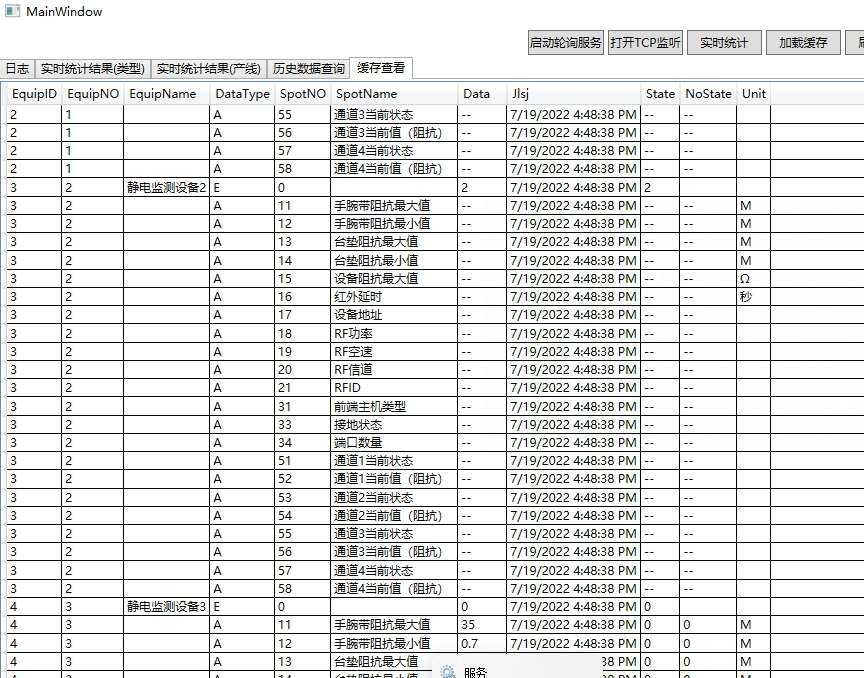
数据包含：产线组名称、产线唯一ID、设备名称、设备唯一ID、设备类型

产线组查询接口：url/SystemSet/GetProductionLineGroupList

请求格式：post、header传递token(传递示例参考图一)

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{“EquipGroupName”:” 产线组名称”,” EquipGroupId”:” 产线唯一ID”，” Equips”:{“EquipNO”:”设备编号”，” EquipName”:”设备名称”,” TypeBinding”:”设备类型”}}}

## 增加读取设备实时数据接口



上图表数据需全部上传。

接口要求：

1. 按设备唯一ID请求指定设备数据

接口：url/Statistics/GetLsjlByEquipID

请求格式：post、header传递token(传递示例参考图一)

参数：{“equipid”:”设备唯一ID”}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{“EquipID”:” 设备表ID”,” EquipNO”:” 设备编号,” EquipName”:” 设备名称,” DataType”:” 数据类型,” SpotNO”:” 测点,” SpotName”:” 测点名称,” Data”:” 数据”,” State”:”状态”}}

1. 按产线唯一ID请求产线下所有设备数据。

接口：url/Statistics/GetLsjlByGroupID

请求格式：post、header传递token(传递示例参考图一)

参数：{“groupid”:” 产线id”}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{“EquipID”:” 设备表ID”,” EquipNO”:” 设备编号,” EquipName”:” 设备名称,” DataType”:” 数据类型,” SpotNO”:” 测点,” SpotName”:” 测点名称,” Data”:”数据”, ” State”:”状态”}}

1. 一次性请求所有数据（是否可以?）

url/Statistics/GetAllLsjl

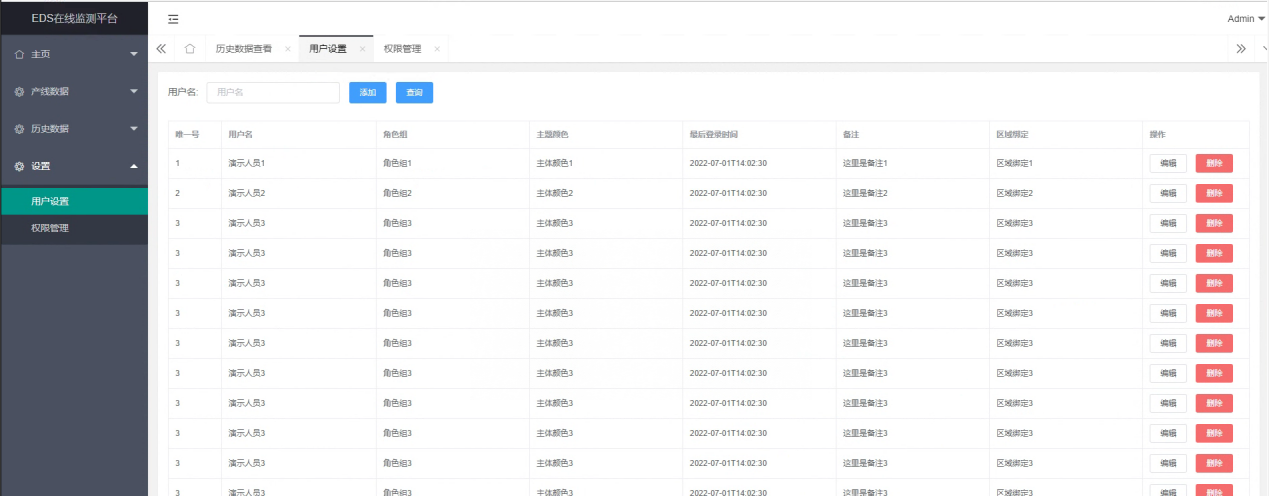
参数：无

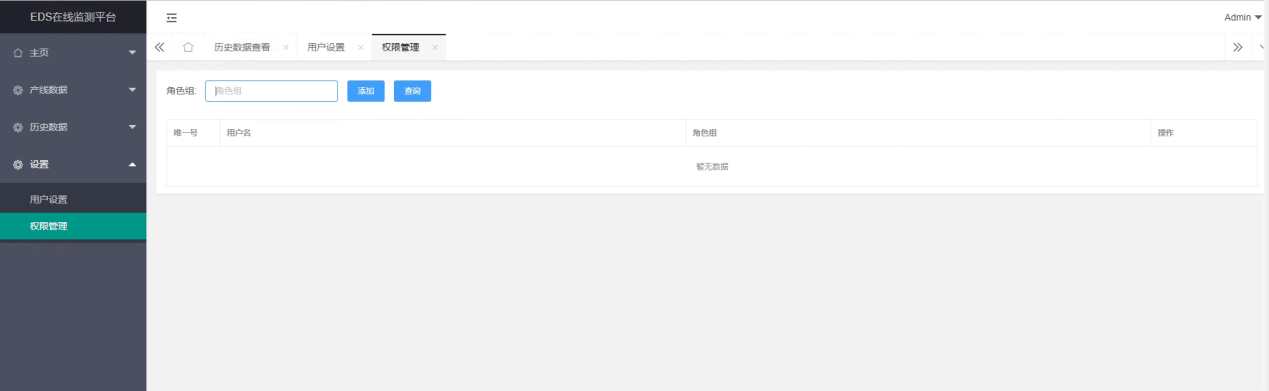
数据结构要包含 、设备唯一编号、设备类型（有？）、设备名称、测点类型、测点编号、测点名称、当前值、刷新时间、测点状态、屏蔽前状态、单位。

请求格式：（补充）

返回格式：（补充）

## 增加用户组配置接口





可添加、删除、修改、人员账号，可对每个角色进行功能授权。

默认Admin账号不可删除，但可修改密码及账号名称等。

密码显示禁止明文，后后台交付需MD5加密传输，

新建账号默认密码888，可修改。除超级管理员外不看查看其它账号密码。

**录入信息需包含以下部分**

1. 用户名
2. 角色组
3. 密码
4. 自定义主题颜色（预留）(使用此用户登录自动切换此颜色)
5. 最后一次登录时间（预留）
6. 备注
7. 区域绑定（预留）

新增人员接口：url/UserManagement/AddUser

请求格式：post

参数：{"username":"Admin123","passWord":"ebpjWHBHhv0YhqDXA4zExA==","roleGroup":"角色组","themeColor":"主题颜色","remarks":"备注","zoneBinding":"区域绑定"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

修改人员接口：url/UserManagement/ModUser

请求格式：post

参数：{"id":4,"username":"Admin123","passWord":"ebpjWHBHhv0YhqDXA4zExA==","roleGroup":"角色组","themeColor":"主题颜色","remarks":"备注","zoneBinding":"区域绑定1"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

删除人员接口：url/UserManagement/DelUser

请求格式：post

参数：{"id":3}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

查询人员列表接口：url/UserManagement/GetUserList

请求格式：post

参数：{“text”:”检索用户名” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”: {

“id”:”唯一号”,“username”:”用户名”,”passWord”:”密码”,”roleGroup”:”角色组”,” themeColor”:”主题颜色”，” remarks”:”备注”,” zoneBinding”:”区域绑定”,” LastLoginTime”:”最后登录时间”}}

### 权限层级说明

1. **展示界面权限**、按菜单项目授权（如：可视化大屏、产线界面、用户管理、系统设置等）
2. **部分页面有设置部分按钮**（按可查看、可控制功能授权）
3. **按产线和产线下绑定设备查看数据授权**（除大屏外的数据、如产线查看、历史数据查看，没有权限的，不能查看此产线和产线下设备实时和历史数据，对应选项也自动隐藏）

产线绑定后，默认没有权限，需要到用户管理里面授权

### 功能说明

可添加、删除、修改、角色，可对每个角色进行功能授权。

角色权限划分

a、主要功能项（一级二级菜单）如：可视化大屏、产线界面、用户管理、系统设置等界面权限。（没有权限、账号登录后，对应菜单不显示，需考虑上下级关系）

b、预留后期页面部分设置按钮授权。仅查看和可设置

c、新建角色默认给所有权限，操作者点击修改后可设置权限。

默认角色分为超级管理员、管理员、操作员，用户可自行新建删除。

1. 其中超级管理员角色不可删除，不可修改权限（所有权限）

新增角色接口：url/UserManagement/AddRole

请求格式：post

参数：{"RoleName":"角色名称" }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

修改人员接口：url/UserManagement/ModRole

请求格式：post

参数：{"RoleId":2,"RoleName":"角色名称22" }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

删除人员接口：url/UserManagement/DelRole

请求格式：post

参数：{"RoleId":2,"RoleName":"角色名称22" }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

查询人员列表接口：url/UserManagement/GetRoleList

请求格式：post

参数：{“text”:”检索角色名” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”: {

{“RoleId”:”角色唯一号”,“RoleName”:” 角色名称” }}

新增功能接口：url/UserManagement/AddFunction

请求格式：post

参数：{"FunctionName":"功能名称"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

修改功能接口：url/UserManagement/ModFunction

请求格式：post

参数：{"FunctionId":2,"FunctionName":"功能名称2"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

删除功能接口：url/UserManagement/DelFunction

请求格式：post

参数：{"FunctionId":2,"FunctionName":"功能名称2"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

查询功能列表接口：url/UserManagement/GetFunctionList

请求格式：post

参数：{“text”:”检索功能名” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”: {

{“FunctionId”:”功能唯一号”,“FunctionName”:” 功能名称” }}

新增角色的功能接口：url/UserManagement/AddRoleFunction

请求格式：post

参数：{"RoleId":2,"FunctionId":1}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

修改角色的功能接口：url/UserManagement/ModRoleFunction

请求格式：post

参数：{"id":2,"RoleId":2,"FunctionId":22}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

删除角色的功能接口：url/UserManagement/DelRoleFunction

请求格式：post

参数：{"id":2,"RoleId":2,"FunctionId":22}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

查询角色的功能列表接口：url/UserManagement/GetRoleFunctionList

请求格式：post

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”: {

{“Id”:”唯一号”,“FunctionId”:”功能唯一号”,“RoleId”:” 角色唯一号”,” RoleName”:”角色名称”,” FunctionName”:”功能名称” }}

通过角色Id查询对应功能列表接口：url/UserManagement/GetRoleFunctionListByRoleID

请求格式：post

参数：{"roleid":"2"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”: {

{“Id”:”唯一号”, “FunctionId”:”功能唯一号”, “RoleId”:” 角色唯一号” ,” RoleName”:”角色名称”,” FunctionName”:”功能名称”}}

## 增加系统设置接口

标题名称自定义修改

Logo自定义替换功能（前端是否可以独立完成？？）

系统设置查询接口：url/SystemSet/GetSystemSettingList

请求格式：post、header传递token(传递示例参考图一)

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{“Title”:” 标题”，”ID”:”唯一号” }}

系统设置新增接口：url/SystemSet/AddSystemSetting

请求格式：post、header传递token(传递示例参考图一)

参数：{"Title":"标题"}

返回格式：{“Status”:”状态”,” Message”:”消息” }

系统设置修改接口：url/SystemSet/ModSystemSetting

请求格式：post、header传递token(传递示例参考图一)

参数：{"Title":"标题","ID":2}

返回格式：{“Status”:”状态”,” Message”:”消息” }

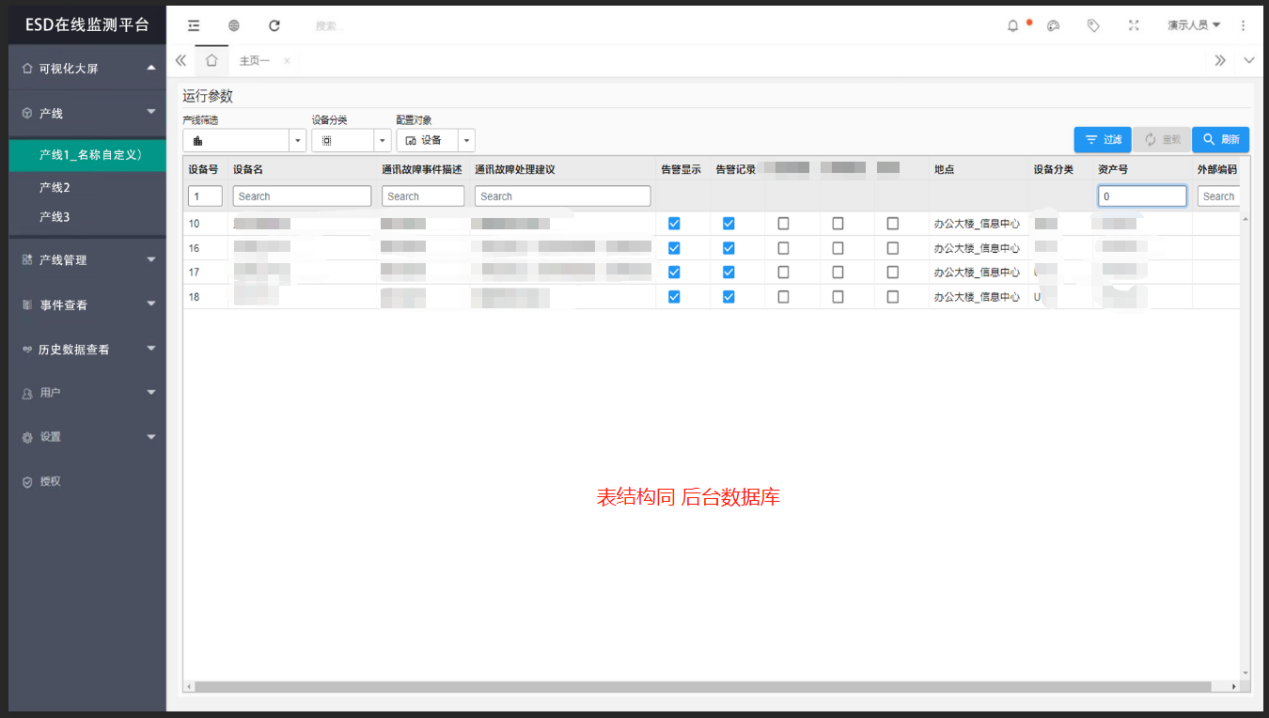
系统设置删除接口：url/SystemSet/DelSystemSetting

请求格式：post、header传递token(传递示例参考图一)

参数：{"Title":"标题","ID":2}

返回格式：{“Status”:”状态”,” Message”:”消息” }

## 增加（运行参数）事件读取接口



* 产线筛选：支持全选、指定产线（产线名称后台定义）、筛选历史数据。
* 设备分类筛选：支持按类型筛选。
* 时间段查询数据（页面没排版）
* 按类型查询数据（数据库E、A、D三种类型）过滤数据。
* 支持针对列表筛选数据（关键字筛选）

接口：url/Statistics/GetEvent

请求格式：post、header传递token(传递示例参考图一)

参数：{"groupid":"1","equiptypeid":"2","ksrq":"2010-10-10","jsrq":"2022-10-10","datatype":"3","eventkey":"4"}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{“GroupName”:” ”,” TypeName”:””,”EquipName”:”设备名称,” DevID”:””,” SpotID”:””,” SpotName”:” 测点名称,” Event”:””,” Proc”:””,” StartTime”:””,” EndTime”:””,” DValue”,””,” Level”:””,” IsConfirm”:””,” Confirmor”:””,” Confirm\_time”:””,” Confirm\_opinion”:””}}

## 用户登录接口调整

现在的接口好像没有用加密吧，验证？ 退出？

登录接口：url/Login/Login

请求格式：post

参数：{“username”:”用户名”,”password”:”MD5加密的密码”}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Token”:”token值” }

刷新Token接口：url/Login/RefreshToken

请求格式：post、header传递token(传递示例参考图一)

参数：{“username”:”用户名”,”password”:”MD5加密的密码”}

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Token”:”token值” }

说明：登录时会签发一个Token值给前台，用于后面处理业务验证使用，但是默认Token的有效期30分钟(可配置），需要前台定时调用刷新Token方法来替换为最新的Token值来进行验证（Token值本身创建后就没法再更新有效期了）

登出接口：url/Login/LoginOut

请求格式：post、header传递token(传递示例参考图一)

参数：{“username”:”用户名” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息” }

## 按设备类型统计实时情况

登录接口：url/statistics/ GetEquipStatusByType

请求格式：post

参数：{“typeid”:”” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{…} }

## 按产线实时情况

登录接口：url/statistics/ GetEquipStatusByGroup

请求格式：post

参数：{“groupid”:”” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{…} }

## 历史曲线左边测点列表

登录接口：url/statistics/ GetLscdlb

请求格式：post

参数：{“groupid”:”” }

返回格式：{“Status”:”状态(true:成功 false:失败)”,” Message”:”消息”，”Data”:{…} }

## 查询曲线

登录接口：url/Statistics/Getlsqx

请求格式：post

参数：{"spotno": "12A54","ksrq": "2022-07-01","jsrq":"2022-07-31"}

返回格式：

{

    "status": **true**,

    "message": **null**,

    "data": [

        {

            "rq": "2022-07-24T16:08:12",

            "data": 11

        },

        {

            "rq": "2022-07-24T23:58:56",

            "data": 12

        }

    ],

    "token": **null**

}

## TCP上报接口

**设备上报数据：** {line＝01,id＝01,qrcode=444444477jjjjj,adc0=3393}

格式错误，不处理。

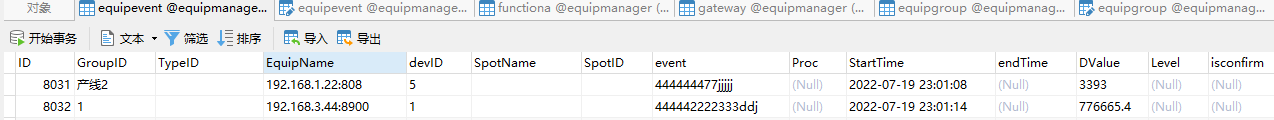
**上位机返回：**收到后返回参数(目前下位机不用，但还是收到返回下吧。)

返回数据前面加一个 OK

{ OK,line＝01,id＝01,qrcode=444444477jjjjj,adc0=3393}

**参数说明**

* Line= 产线ID： 如上报过来ID和equipgroup 表的EquipGroupId相等，就填写对应产线名称（EquipGroupName），将产线名称 写入 equipevent表的 GroupID字段。 如对应ID产线不存在，就直接写入line的数值。
* id=设备ID：不用转换，直接值写入 devID 设备ID
* qrcode= 二维码信息 ,写入equipevent表的event字段（直接写入）
* adc0=当前值， 写入equipevent表的 DValue字段。（值可能要做转换，待定）
* StartTime 开始时间，写入时，写入对应接收到时的北京时间。
* EquipName 设备名称写入上报过滤的对方IP和端口。



## 控制接口

**与网关请求接口完整格式:**

192.168.1.16/cgi-bin/webctrldev.cgi?device\_id=61&cmd=06&cmdparam=1392&param\_value=32&timeout=&level=&IsCurve=&cmd\_notes=开机

网关返回值为[{"result": "ok"}]代表成功。

* device\_id：设备号， 对应Equip表里面的EquipNO（注意是某个网关那边的设备号）
* cmd= 命令，前端写入
* cmdparam：命令参数 ，前端写入
* param\_value:值，前端写入
* timeout=&level=&IsCurve= 可为空，前端写入
* cmd\_notes=说明

1. 需要消息队列，缓存处理。（前端可能会批量调用接口，写入设置值。后台缓存下来依次传网关执行）
2. 前端调用接口：可以批量调用，不要一个一个调用。

后台返回接口：返回成功或失败结果。

1. 需要写入日志，（数据库新加日志表。）LogEvent

* Type 日志类型： （数值型）
  + - 1：系统登录日志
    - 2：后台启停日志
    - 4：修改系统设置日志
    - 8：修改设备参数日志
* GroupID 产线ID（数值型）
* Event 内容（文本型）
* user 操作用户（文本型）
* StartTime 操作时间
* Ipaddr 操作IP和端口（是否可以记录操作者IP）

**调用接口成功写入日志内容如：**

Type 8

GroupID 6

Event 设备ID对应名称；cmd\_notes ;cmd; cmdparam; param\_value

**Event完整事件内容如：**离子风机设备6/6-152； 控制开机；06；0033；33.5

user Admin

StartTime 2022-07-30 20:31:00

Ipaddr 192.168.2.1:80

## 事件记录及内部联动接口



**事件保存功能优化：**

1. 原来是采集数据变化就存储一次，启动时全部存储一次。现在修改为满足指定状态条件才存储。（有的配置可能还要独立配置，如温度高温报警值不一定每个都是30）

需要满足至少多种种情况产生报警，同时记录事件。

1. 当值大于或小于某个值触发报警。（如温度高于30度触发高温报警，低于28度恢复高温报警，低于5度触发低温报警，高于10度恢复低温报警。，同时指定报警触发级别，后期通过级别过滤报警输出等功能要用）（要做）
2. 当测点在对应值触发报警。（如=1代表开机 ；=2代表关机；=3代表待机。同时指定报警触发级别，后期通过级别过滤报警输出等功能要用）（要做）
3. 当测点变化过大产生报警。（如测点一下变化超过百分之5，立刻触发变化量过大报警，同时指定报警触发级别，后期通过级别过滤报警输出等功能要用）（要做）
4. 当测点长时间没变化产生报警。（如测点联系多久一直没变化，也触发报警，同时指定报警触发级别，后期通过级别过滤报警输出等功能要用）（预留功能）
5. 告警升级功能，当测点一直在报警，但一直没处理，满足配置时间2小时，报警升级，再次触发一次报警，同时指定报警触发级别，后期通过级别过滤报警输出等功能要用（预留待定）

**告警屏蔽功能。对象为设备和测点（屏蔽后此测点不产生报警）**

屏蔽分为设备屏蔽和测点屏蔽，设备屏蔽后此设备下所有测点都不触发报警了。测点屏蔽，就这个测点不触发报警了

## 新加告警联动功能。

当前面“事件记录及内部联动接口”产生事件时，同时看此测点是否有配置联动接口，要有配置，报警时触发联动功能。恢复时也要触发联动。

面向对象不是类型设备了。是整个系统每个采集测点。

## 历史数据保存功能优化：

**增加变换率存储+间隔时间存储和 整定存储2组合策略。**

**变换率存储+间隔时间存储：** 当要存储测点满足2个策略条件才存储一次记录，从而优化数据存储。

如：测点配置百分之0.5变换率，600秒最长间隔存储时间，上一次采集值为53400，下一次采集回来为38000，满足百分之0.5变化率则立刻存储一次数据。

如下一次采集回来为53410，不满足百分之0.5变化率则不存储。如一直不满足变换率，但和上次存储时间相隔大于配置的10分钟（600秒），就也存储一次当前采集数据。（这里的上一次是本次采集和前一次存储值对比，）

注意1：间隔时间范围（1秒-86400秒）；如配置错误，间隔时间是0秒或1秒或没有配置，就按1秒一次存储。如间隔时间大于1天，就按1天存储。

注意2：变换率范围：最小0%--最大100%,精确到0.01% (如配置错误，变换率配置小于百分之0 ，就按0处理）。

**整定存储：**为每半小时或每1小时存储一次。和前面间隔时间差不多，只是必须如12点30分或1点整才存储，其余时间不存储数据。

注意3：**变换率存储+间隔时间存储 和 整定存储 只能2选1.**