|  |
| --- |
| **内部资料 严禁外传** |

# 

# **基于网关**

# **数控引擎概要设计**

**（v1.0.0）**

项目编号：QX-IOT-2020.11.11

**战略信息中心 软件开发部**

**2020年11月11日**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本编号 | 修订模式 | 修订内容 | 修订人 | 修订时间 |
| **v1.0.0** | **A** |  | **肖晓颖** | **20201111** |
|  |  |  |  |  |
|  |  |  |  |  |

文档修订记录

\*修订模式：A-新增、D-删除、M-修改

# 引言

# 编写目的

为明确软件需求、安排项目规划与进度、组织软件开发与测试，撰写本文档。

本文档供项目经理、设计人员、开发人员参考。

# 项目背景

基于千玺研发的多种设备，每个设备都 对应一套调度的程序来驱动设备，对人力物力损耗比较大的纵向开发模式；

# 定义

# 参考资料

# 任务概述

# 目的

1. 为了解决对与纵向本地调度程序；
2. 为了解决以后设备应用从开发过度到配置化的应用程序；
3. 为了解决兼容多设备的调度执行；
4. 为了解决设备更新换代后可以从配置中适应；

# 运行环境

# 硬件环境

* 本地终端中控机
* 本地终端服务器
* 局域网内

# 软件环境

* Centos
* Mysql
* Redis

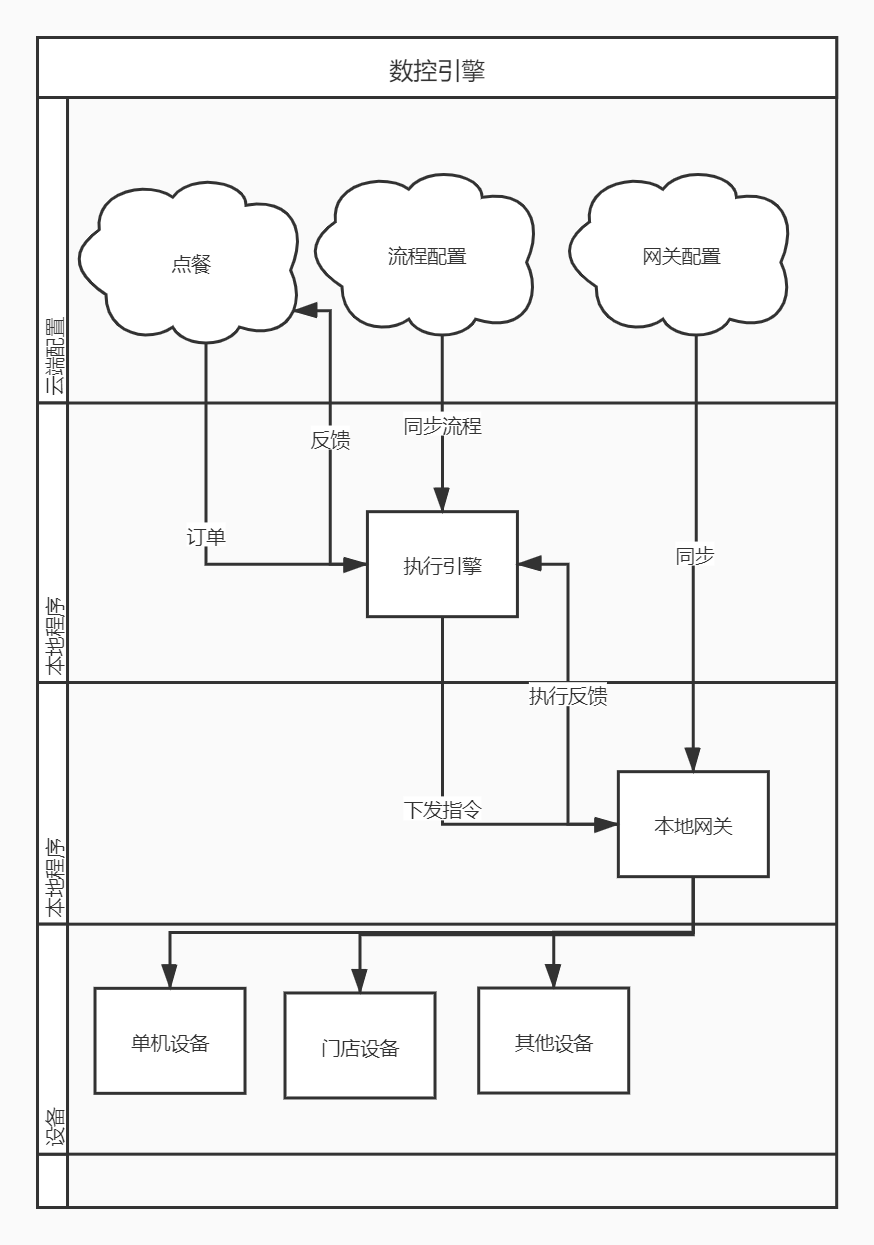
# 条件与限制

# 数据描述

# 数据类型

1. 流程数据（xml)
2. 配置数据（数据库）
3. 设备采集数据(数据块）
4. 接口数据（json)

# 数据流向规划



# 功能需求概要

# 功能划分

前端程序

1. 网关配置
2. 工艺流程图绘画

云端程序

1. 配置获取接口
2. 配置数据管理

本地程序

1. 同步配置
2. 订单执行/反馈
3. 设备点检
4. 数据采集/报警

# 功能描述



# 云端配置

# 数据结构管理

数据块配置，是为了每个设备的数据可以成块读取。如：一个short的类型可以分为16bit类型的小块；又例如几个类型基础类型可以组成大的数据库；

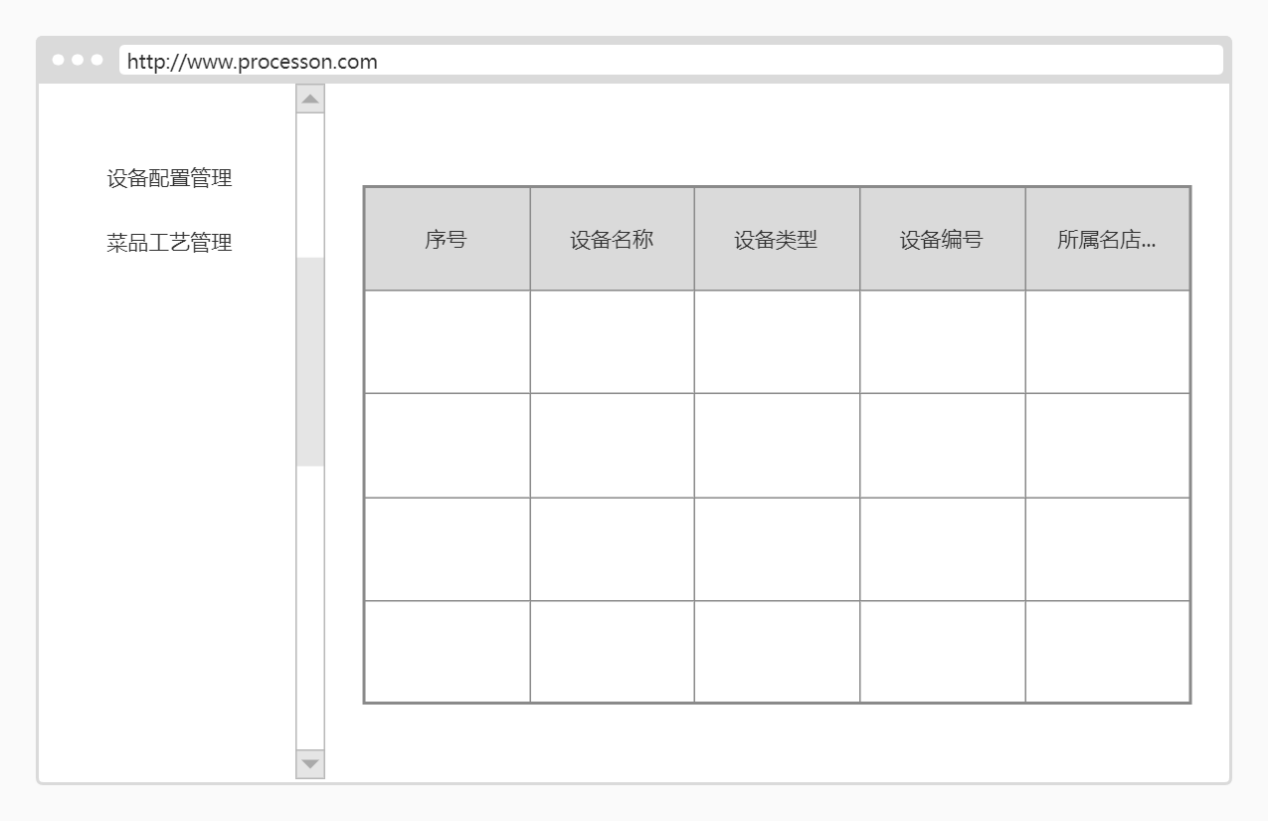
在网关里面 json 转义成数据块 或者 数据库转义成json 的数据结构，都需要配置的支撑；

配置界面：



数据结构：

# 配置管理



这里针对每个一个设备进行配置；配置设备的ip / ID /io /方法等；每一行配置都对应一个设备；

# 方法功能

# 属性（io点）

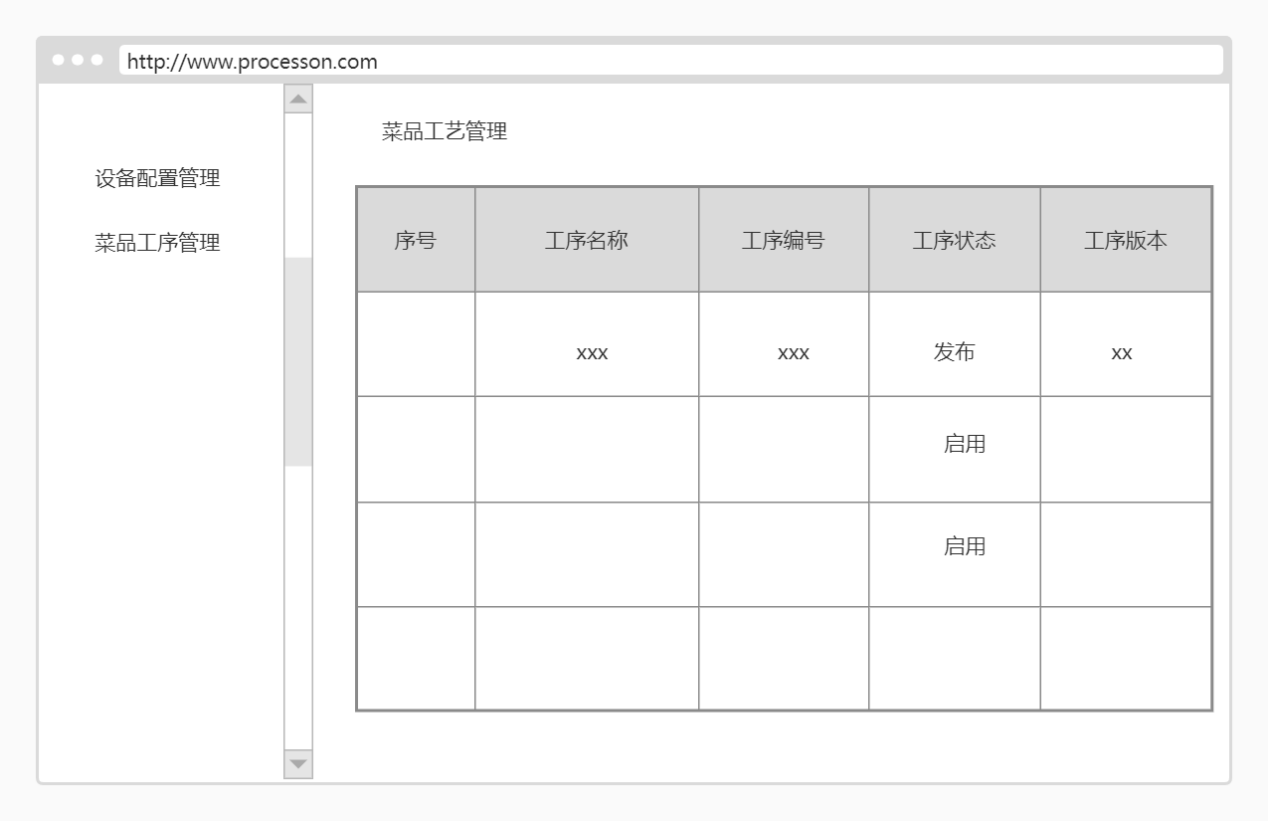
# 报警

# 执行反馈



# 调试功能

# 工艺流程配置管理



# 流程发布/禁用

# 流程添加

# 流程修改

# 流程删除

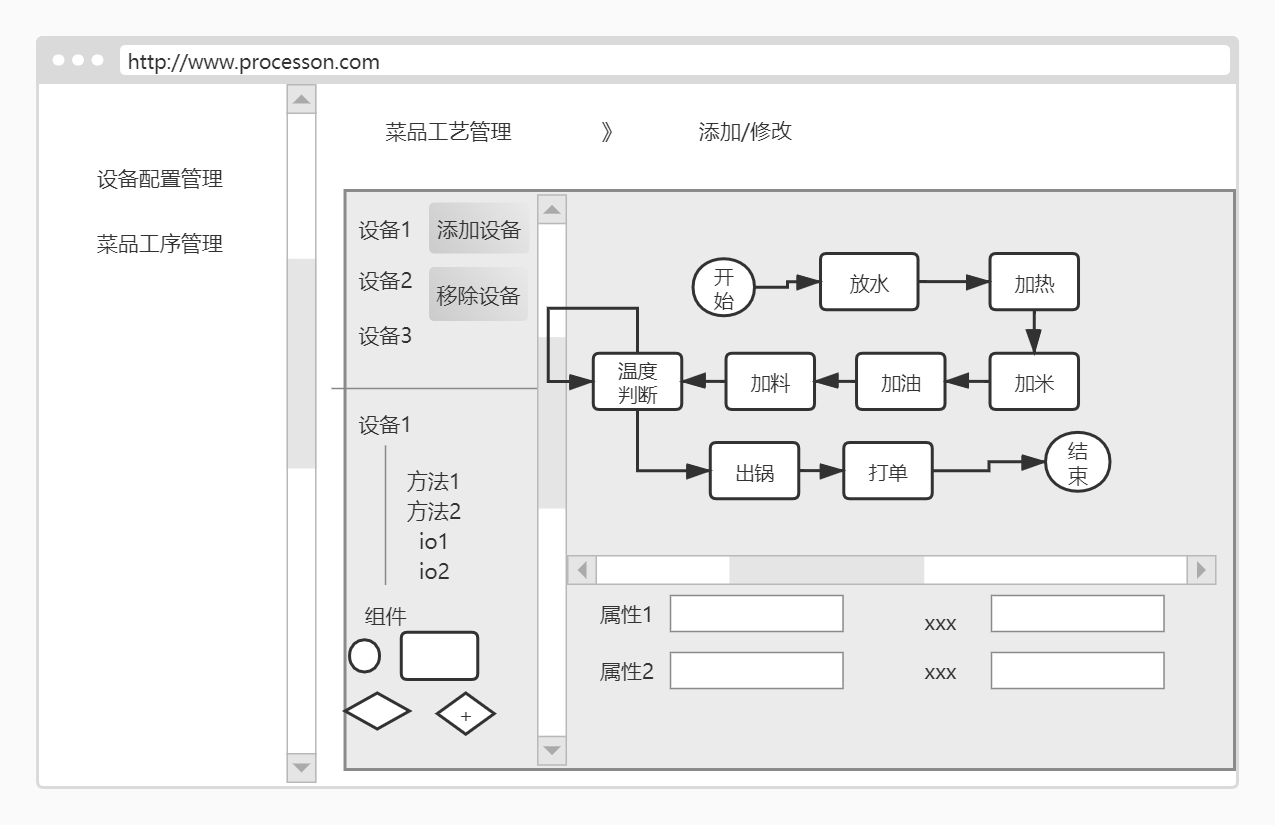
# 流程复制

# 节点管理（开发）

# 节点发布/禁用

# 节点删除

# 节点修改



# 门店设备管理

# 设备管理

一个门店下有多个设备；

# 执行日志

每个设备的执行日志；

# 单机设备管理

# 设备管理

每个单机设备管理；

# 执行日志

每个单机设备的执行日志；

# 设备类型管理

同一个类型的配置可以复用，那在配置一个设备的时候，可以选择类型设备进行快速配置；

# 本地程序

# 设备点检

# 数据采集

# 调用网关

# 点检执行

# 设备报警

# 本地调度

# 流程同步

流程同步模块

# 执行引擎

根据订单分解后的发布的流程进行执行模块

# 日志管理

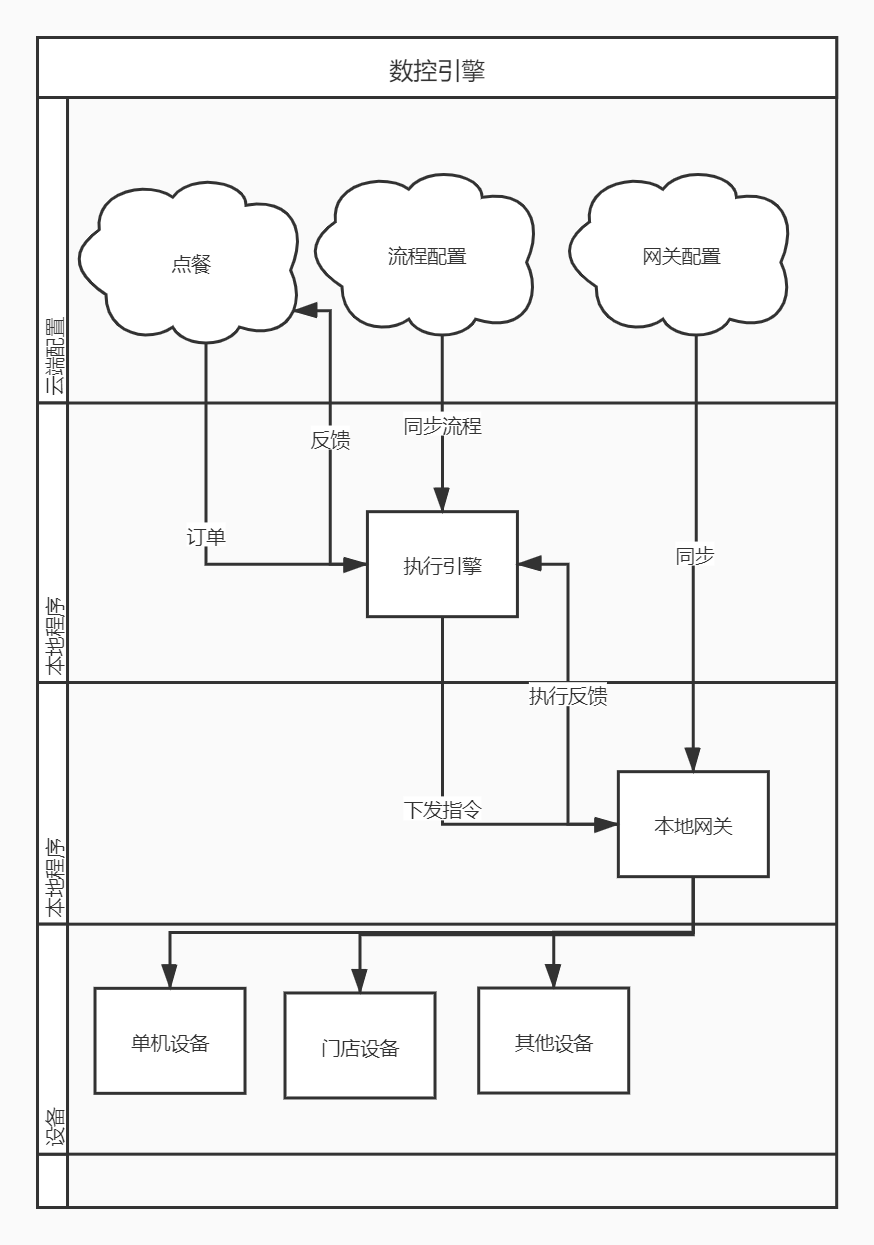
执行日志

# 调用接口

集成即可，给订单系统使用

# 调用网关

对网关接口进行调用



# 本地网关

# 调用接口

给外部程序调用的接口

# 网关配置同步

同步云端程序

# 网关核心程序

根据配置，对设备进行数据采集命令执行

# 日志管理

执行日志

# 流程图案例描述

