接活太困难了。

设计一个自己的工作管理平台，很多事情能够逐层的被管理和分派，任务最终被确定执行

以岗位和任务为基准，能够对岗位和任务进行细分。

对于一个人或者职位，要求有确定的输出验证

拥有不同的权限管理

项目名：7512

整个流程以任务的方式运行，用户可以报名参加。操作单元：输入--流程--输出

上家确定输出，下家确定输入和流程

情景：

雇主A：我要做个应用，谁帮我

工人B：概念设计->界面设计->交互设计->产品开发->交付

工人C 工人D 工人E 工人F

工人C：市场调研->用户分析->功能确定

功能设计：

每个用户可以建立若干个空间，每个空间可以有若干个起始节点，每个节点可以有若干个前置节点和若干个后置节点，并且每个节点也可以拥有一个子空间。

用户能够编辑节点的输入条件和输出条件，以及对节点的称呼和描述，能够连接多个节点，还可以依据节点中的信息推荐合适的人。节点拥有若干可以自定义的扩展属性。

用户也拥有若干属性可以用于筛选搜索和推荐

1.0版数据库设计

节点 后继节点列表

节点属性

节点子空间起始列表

用户 用户属性

api设计

对某一节点设定/添加前置节点

对某一节点设定/添加后继节点

在某两个节点之间插入一个节点

删除某个节点

添加/修改/删除某个节点的某个属性

在节点子空间中添加一个起始节点

设计原则：代码复用率最高？网络流量最小？服务器负载最低？

以代码复用率最高为原则。

客户端只负责显示数据和将动作传递个服务端，如果某个界面需要多种请求，有客户端拼接，可以最大限度减少api的数量

服务器端负责响应动作，处理数据，提供原始数据，不对数据做显示的加工。

服务器端相当于业务逻辑层+数据层，客户端相当于显示层

当有个别严重影响的多次调用接口，服务器则提供一个优化版的一次性处理接口

接口分为两部分：核心接口和快捷接口

核心接口：//基础核心接口

设定前置后继节点（前置节点标识，后继节点标识）

创建节点（节点数据）：节点标识

删除节点（节点标识）

修改节点属性（节点标识，属性名，属性值）

添加子空间起始节点（母节点标识，子节点标识）

修改用户属性（用户标识，属性名，属性值）

快捷接口：//核心接口的封装