**企业版总结**

需求：

1. 部分数据库查询过慢，原有数据库搜索引擎无法承受，有诸多统计型需求无法添加
2. 代码结构属于从上到下，复制粘贴型的，废代码太多，不好维护
3. 移动端和PC端程序项目不统一，本来两者的逻辑上应该无太大差别，只是细小的参数和返回项不统一而已，结合在一起可以复制相同的一套后台设置，进行简单差异化修改即可完成后台设置，而且这种方式添加和修改展示项可以避免频繁的发布版本
4. 支持多语言

概述：

1. 框架结构：apiz1(专业版接口)、apiv1(移动端接口)、common(公共程序)这三个项目文件，后面新增的项目文件都去继承common项目的内容，apiv1/apiz1功能无差异时无需重写
2. 采用技术和模式：

程序采用简单数据库使用MYSQL进行后台设置即可完成上库，复杂数据库采用ES,前端通用的展示项通过后台配置，配置完成之后，需要程序添加相应的查询逻辑即可完成。(注:Elasticsearch 是一个分布式、可扩展、实时的搜索与数据分析引擎，它最大的特点是能够快速的进行聚合统计，其次才是查询，但是他在创建索引维护数据时繁琐，虽然已经封装，但是在写法上任然会与mysql的写法有出入，所以采用的是两种模式。一般操作都是用的什么搜索引擎就调用谁的model实例,但是会有像扩展信息这种，本项目需要查其它表（es/mysql），这个时候就新建了一个config model的实例方法，自动切换)

前端与后端采用调用接口的形式，前后端完全分离。这种方式好处在于，各干各的事，双方都不用去填充别人的东西，但是坏处就是双方需要勾兑下格式，还有就是我们的展示都是后台配置的，会经常提一些问题过来，有些是后端没返回，有些是前端没有加，搞不清楚就以为是设置的问题，其实最简单的方式就是看请求数据返回该项没有，如果有就是前端有问题，如果没有就是后端有问题

redis 缓存 在用户输入搜索记录时，一段时间内，如果重复提交只记录一次

推送是极光推送

1. 后台使用及注意事项

后台添加数据库时，默认是单表，如果是复杂数据库请先联系开发人员，让他先去添加，然后再进行相应的设置和开发操作

产品设计时，为了防止误删除，所以删除都是做的软删除(只是修改了状态),数据实际上还是保存在数据库里，所以在添加的时候请想清楚再添加，不要随便乱添加

在进行扩展信息设置时，参数和字段对应的条数要一一对应，和老版的设置方式一致，在设置pian和jinkou这种上下关系链接的时候关联表设置用&链接,例:pian&jinkou

设置的扩展信息关联过去的查询有时会包含一些没有设定过的参数，这个时候需要隐藏参数，但是像老版本设置那样直接选择，前端那边有冲突，不好处理，需要单独进行隐藏参数的设置

上传下拉选项时会全部覆盖，不会一个个的去判断做增量添加

要添加特殊字段权限时，请先勾选设置为特殊权限，然后再打开相应的数据库进行设置，权限设置的正确顺序是 设定是否特殊->设置哪些数据库->设置数据库权限

会员登录先判断yaozh的登录，再进行相应的项目登录。（登录不上去的大部分属于没有到项目里面去添加账号）

项目反思：

理想和现实往往是有差距的，差距还不是一丁点，各方面的原因都有。大部分人都是后面慢慢加进来的，对项目的逻辑和老版数据库的认知有的有一定的基础，有的是基本不清楚，摸着石头说一点做一点，没有连贯性。设置方面为了提高性能，把原来在程序中拆分的部分在后台添加的时候进行拆分，这样关联着修改和删除的时候就需要很麻烦的去判断原来的默认，经常容易改掉，还有就是后台模板在进行下拉选项值选中时，没有默认值会报错，稍不注意就改掉一个，还有设置的时候不知道规则进行了错误的设置，等等问题。如果下一次项目开发的话，我觉得应该是先定下人，每个岗位涉及到的人先定下来，约定规则，开发一个各项基本功能完善的版本，然后在讲解给相关的开发人员依样画葫芦。然后有人会提出来后面来的人得不到锻炼，其实非也，每个人的成长是都是先学习而不是先创新，你进来看到的都是前人的经验和不足，对自己来说成长比去花大把的时间创新来得快，到下次项目开发的时候你也积累了不少的经验了，这个时候可以参与下个项目的初始开发，这是我认为一个比较好的进阶模式。