

WeRun后端24暑期纳新第二阶段

首先，恭喜大家完成了第一次作业，想必尤其是 Java 方向的同学会觉得本次作业的难度不是特别大，业务也不是相当复杂，这也旨在让大家进行练手与学习～

第二阶段的任务有两个：

- 将第一阶段的后台服务进行部署上线，与前端对接接口。我们将在 8.15 左右进行，请同学们关注群内通知。[📖 对接工作说明](#)
- 我们接下来要完成 "**项目方案设计书**"，具体要求如下：

具体实施方案

在实际开发的过程中，常常会出现一些意想不到的问题，其中一些问题值得我们思考与回味。

下面有五个实际开发当中出现的问题和困难，请同学们对下面的问题**五选一**，写一个项目设计书，无需实现完整而又具体的代码，需要根据题目的情景进行可行性设计，说明：

- 对需求的分析
- 设计的项目结构
- 运用的技术栈
- 关键技术栈的关键代码
- 可以写一个小 demo 进行展示。

问题列表

项目1: WeRun 官网 - 注册登录

作者 张沅睿 负责人 王子青

最近，WeRun前端团队在刁👉的设计与彭于晏同学的推动下，正在开发一套**实验室官网系统（official-site）**，用于宣传实验室吉祥物朱ger，顺便再带上前端纳新。



方向



正在开发中
非最终效果
仅供演示

项目



通过HTML、CSS、JavaScript
构建Web应用

毕业去向



加入我们

加入我们

现在这套系统正在面临两个问题。

问题1：该系统部署在公网，在互联网上任何人都可见。不过在注册登录时，需要验证新注册的用户是否为哈工大校内学生。但是我们**没有权限掌握校内所有学生的数据库信息，也无法接入哈工大的统一身份认证系统**。请提出一种方案，尽可能平衡对于**实验室管理员与纳新候选人**的方便程度，并且尽可能保证注册成功的同学确实是我们校内的学生，也尽可能保证安全性，对于访问凭证泄露做好防备。

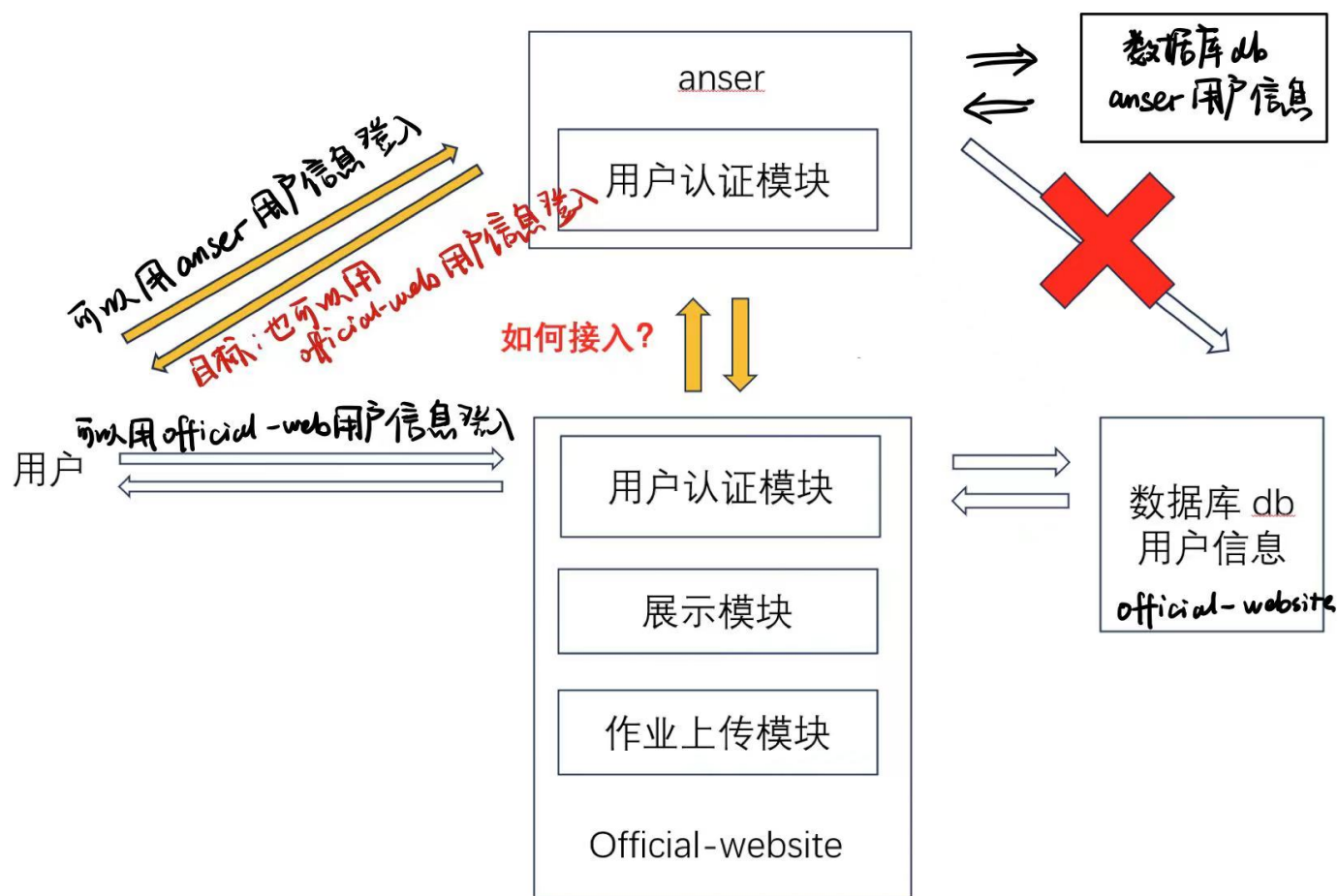
（提示：这个问题可以尽情打开思路，不止限于技术手段！我们有纳新QQ群和学生邮箱，这些都可以利用起来）



问题2：很不巧，WeRun后端也开发了一套纳新系统（anser），但是由于信息沟通不畅，两个端分开开发好长一段时间才知道对方也在开发，导致所用**技术栈大不相同，且使用了两个数据库**，因此也不可能合并成一个项目，也就是说**两个项目会分开部署，分为两个不同的微服务系统**。

但是对于用户而言，肯定不希望WeRun前后端两个系统是分开登录的（就像登陆完淘宝之后，天猫也是可以用一个账号登录的）。假设问题1此时已经得到解决，WeRun前端的用户系统已经完备，那么**WeRun后端开发的新系统该如何接入WeRun前端的用户认证系统？**请针对 official-site 和 anser 这两个系统如何进行设计，提出一个安全并可行的方案。

注：为了安全起见 anser 系统不能直接读取 official-site 的数据库，即 anser 无法直接使用 official-site 的数据库用户信息进行登录。



项目2: 电梯维保 - 维保计划

作者 刘硕 负责人 刘硕

电梯在我们日常生活中经常用到，它们每天如核动力牛马般载客，任劳任怨（当然哪天怨了你就要寄了），因此我们需要定期派专业人员给电梯进行维护保养。我们有这样一个想法：为每台电梯制定一个**维保计划**。（对于一台电梯要么是常规维保，要么是按需维保，这取决于它的硬件设备）。下面是维保计划逻辑说明文档：

维保计划逻辑说明文档

规则：lsof-ti:2400

维保模式分常规维保模式和按需维保模式。


常规维保模式：根据预先设定的时间间隔来进行，比如半月、季度、半年、年度等，属于定期维保，理论上可以全面覆盖检查、排除隐患、降低风险，保证电梯的正常运行。

按需维保模式：通过物联网设备获取电梯运行参数。根据电梯的实际需要来进行维保，针对性强，能够大幅度减少维保的频次，降低成本，节省社会资源。

常规维保模式的维保项目类型包含有：**半月维保、季度维保、半年维保、年度维保。**

按需维保模式的维保项目类型包含有：**按需维保、季度维保、半年维保、年度维保。**

半月维保：每1-15天时间周期内要做一次的基础维保。

 维保计划逻辑说明文档.pdf

请你仔细阅读上述文档，**准确理解业务逻辑，实现这个业务。**

（需要说明设计数据库表、使用的关键技术栈，画出准确与合适的业务逻辑实现图，并进行实现。）

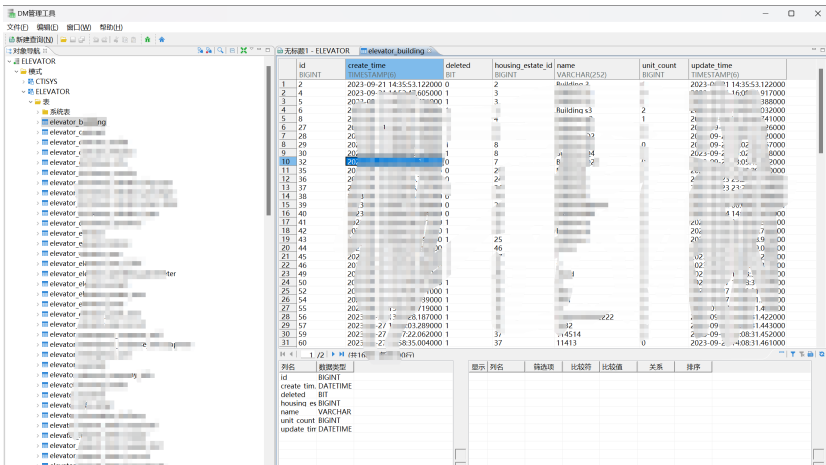
项目3: 电梯维保 - 管理员备份达梦数据库

作者 张沅睿 负责人 张沅睿

2023年的深冬，还是小部员的我突然接到了  魏鹏宇 的一通电话，给我派了一个让我头大的需求：

电梯管理系统的整体是 SpringCloud 微服务框架开发的系统，所用技术栈也比较普适性，但是为了国产化信息保密需要，其所用的数据库可不一般——**达梦数据库**，一款信创的国产操作关系型数据库，相比通常的数据库有几个痛点：

- 不支持 Navicat、Datagrip 等数据库管理软件，必须用它自己自带的 DM数据库 管理工具。
- Jdbc 驱动很刁钻，必须要选好版本号
- 在2023年11月之前不支持 Seata，无法用它实现分布式事务
- 互联网上资料很少，所需要的功能在互联网上可能查不到，遇到 bug 较难排查。
- 导入导出较为麻烦，尤其是从其他数据库迁移到dm数据库时。



现在甲方提了这么一个需求：我们需要该系统的**管理员在该系统的前端页面点击按钮**，可以得到**数据库的.sql文件以进行备份**，以防重要的数据库信息因为各种因素丢失。

也就是说，对于后端开发而言，我们只需要实现一个接口，前端可以通过调接口实现导入导出操作即可。

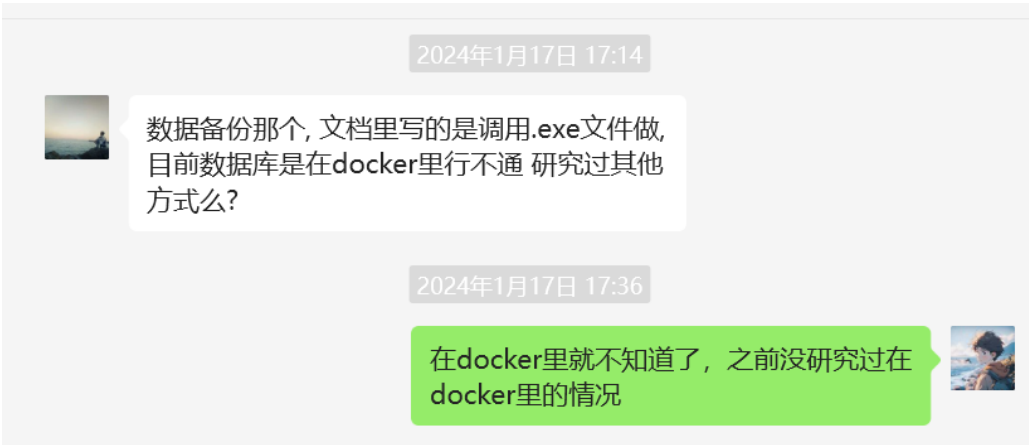
通过**查阅官方文档和使用手册**，我发现我可以通过**使用命令行的方式**对DM数据库进行导入导出，那么问题就简单了：我只需要使用Java代码运行命令行，然后再使用Java做一个Springboot Web程序，就可以实现了！

不只是Java，Golang（Gin）、Python（Flask等）其实也可以轻松做到Web程序，甚至比Java还要容易。不过我那时候只会Java所以在此也只有Java.....

于是我就这样兴致勃勃的写了一篇文档：

[通过调接口实现导入导出达梦数据库.pdf](#)

本以为这样就可以完成任务而邀功的我，就这样把这份解决方案给甲方发了过去，然而事情却远远没有我想的那么简单.....



由于当时我不会 docker，甚至不知道 docker 是什么，于是这个机会一下子就错过了~
请针对这个问题以及下面两个加分项**提出一个解决方案**，弥补一下我当时的遗憾吧~

问题省流：达梦数据库是国产操作数据库，通过查阅官方操作手册可以知道其备份命令，但现在他们使用Docker进行了部署，无法在宿主机直接使用shell脚本；请提出一个解决方案，将其备份的.sql文件可以放到某个目录下，使得用户**在前端页面能够点击按钮直接下载**。

加分项1：用户可以进行**定时任务操作**，将数据库文件自动化定期备份到服务器内。

加分项2：众所周知，肯定不是任何一个用户都能拿到数据库文件的，那样无疑是危险行为！我们要对**这个接口**进行权限设置，保证只有超级管理员角色才能成功访问这个接口。

加分项2.5：实际上，电梯维保这个微服务主系统已经存在了一个 Auth 模块用于用户的权限认证，那么我们如何进行设计，能够通过解析用户发请求携带的token 直接在数据库备份接口进行鉴权呢？

项目4: 意语 - 自动化部署与测试

作者 张沅睿 负责人 张沅睿

随着意语的业务不断扩大，我们发现仅一个人已经没有精力开发所有内容了，于是我们开始了分工开发，开始使用了我们团队合作中最重要的一个工具：Git。

很荣幸，你成为了这个项目的最大牛马 —— **项目负责人**。你需要**配置 Git 仓库**，合理分配大家的任务，将大家提交的代码进行合并，统计每个人的工作量，然后把它部署到服务器上，好让前端抓紧进行测试。

随着开发的人员越来越多，每隔一会就有人找你合并代码，然后前端和甲方让你抓紧把代码部署。最严重的一次是，你的午觉睡着正香，突然收到了一个成员的电话轰炸 —— 他要急着往仓库上推代码，然后叫你赶紧部署，甲方在催！

于是你不得不起床，**打开电脑，打开IDEA，打开Gitlab，合并他的分支，git pull更新代码，打包，部署……**就当你做完这一切的时候，前端又给你打电话：这个接口还是有bug，再部署一下！你逐渐的变的越来越崩溃，气的背地里大骂为什么没有做好测试……

这时候，一股神秘力量突然让你的浏览器里突然弹出了这两个网页：

- 在意语这个项目中集成一个SDK：[GitHub - bearslyricattack/Andulir: 极简,协同的接口自动化测试工具](#)
- 在公网服务器上开一个自动化部署工具：[【转载改编】自动化部署之Jenkins - 扎西拉姆Donnie](#)

请利用以上两篇文章的内容，设计一个方案，使用自动化部署与接口测试工具，拯救濒临崩溃的你免于水火之中吧~

要求：

- 以上两篇文章均为设计思路，你可以选用其它的关键技术栈实现，但是测试必须使用类似Andulir的集成SDK方式。
- 做到在自动化部署的同时进行接口测试。**
- 在微服务系统中，若只更改某一个模块/服务的代码，则**只对该模块的代码进行构建并推送，而不要对已经部署好的模块重新再部署。**

加分项：

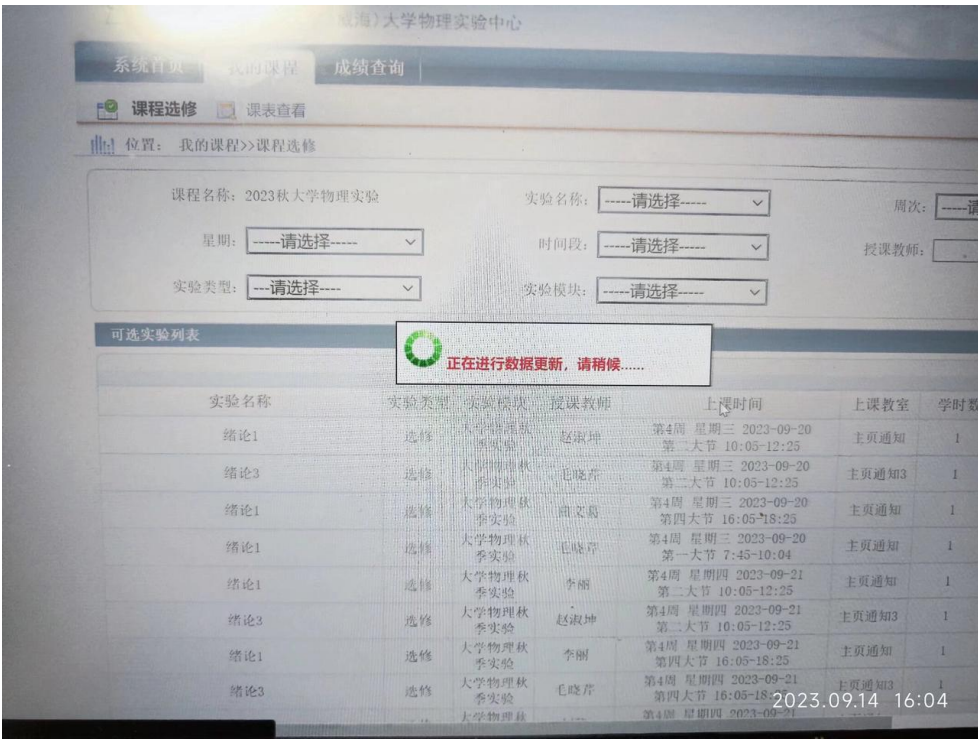
- Jenkins 很重，如果服务器配置不好，就会经常在部署当中宕机，而一旦宕机就会重新启动服务器，很麻烦。考虑使用更轻量级的部署工具，尽量减轻服务器开销。
- 对 Andulir 这个开源项目提出 issue 和 pull request ，谢谢！

项目5: 意语 - 移动端APP

作者 王泽宇 负责人 王泽宇

假如意语的业务经过不断的开发，为了更好的用户体验，我们终于开发出来了一个移动端手机APP。每次当我们打开app的时候，我们需要检测很多信息，比如app的版本号，比如在用户登录之后的 token，比如，一些公用的配置信息，因此，我们在使用app的时候，意味着，我们需要大量调用某一个或者某几个接口。

因此，对于这些接口，我们对其可靠性和处理速度的要求是非常高的，我们设想一下，如果我们在打开app的过程中，经常出现用户登录失败或者app更新检测失败的弹窗，那么对于app使用者来说使用体验一定是非常差的：



- 那么我们基于这个现象提出以下问题。
- 问题1：**如何衡量这些接口的具体性能？（使用什么工具？如何查看？）；如何优化这个具体的性能？（从哪几个方面？）提出一个解决方案。（对方案的合理性进行阐述）
- 问题2：**显然，对于不同的信息，我们使用的频繁程度是不同的，那么我们是否可以对这一现象进行改进，进一步提升性能。（基于问题1 进行改进）

问题3：假设我们的服务部署在多台服务器上，如果不加以限制，很可能出现大量用户请求只访问一台服务器上的接口而不访问其他服务器，造成其他服务器资源的闲置，我们如何去优化这一问题呢？（阐述原理）

以下两个问题的负责人暂为 张津睿

问题4：暂时抛开问题3，假如我们开发一个新业务：发布新版本后，率先更新后的用户可以得到5天VIP资格，但是该资格只有1000份。那么当大量用户使用我们的app时，系统是在多线程高并发地读取内存或者关系数据库，那么当那么多用户使用的时候，我们如何去更新我们的信息，使得VIP资格不会被抢超，同时尽可能让用户有良好的体验呢？（提示：从读写锁方面考虑）

问题4加分项：考虑问题3，如果上述系统是分布式系统，又该如何设计呢？

tips:在我们实际开发中，除了mysql,oracle等关系型数据库，我们还有以kv键值对的形式存储的内存数据库，我们从内存中读取访问的数据远高于从磁盘中读取访问数据速度。

注意事项

- 大家针对各个项目的任何问题，都可以对该项目的出题人和负责人进行提问
- 项目计划书可以参照参考 [目项目计划书 - 示例](#) 来，也可以自己自由发挥，能够说清楚就可以
- 我们会在面试时着重提问大家的解决方案，**一些关键部分会提问到具体实现，并可能需要现场展示 demo**，所以不用想着跑一遍 GPT 就直接蒙混过关了~
- 将以下内容打包命名为 学号-姓名-第二阶段作业-方向 发送至上次作业发送的邮箱 werun_backend@163.com。需要提交的内容有：
 - 项目方案设计书
 - 意语项目部署成功的截图（我们直接向前端询问反馈，不需要大家单独提交了）
 - 个人的简历，便于我们提前了解大家，可以写一些项目经历等等，**建议找一个合适的简历模板**
- 我们会根据 【项目设计书的质量】 与 【一阶段作业与前端对接情况】 确定面试名单。
- 时间安排：
 - 由于前端作业截止日期较晚，我们将在 8.15 左右开始与前端的对接工作，具体需要等待群内通知。
 - 所有提交内容截止日期为：开学第一周周末 9月1日
 - 与前端对接任务同样截止到 9月1日