

个人简历

• 基本信息

姓 名	王盛开	年 龄	27 岁
性 别	男	籍 贯	辽宁省本溪市
民 族	汉族	联系电话	15901296447 18202417075
专 业	机械设计制造及其自动化（液压）	学 历	本科
现居地址	北京市昌平区天通苑北二区 14 号楼	E - Mail	18202417075@163.com skwangdlkepler@gmail.com

• 求职意向

工作岗位	Java 或 Java 全栈
工作地区	北京
工作性质	全职
工作年限	2.5 年
薪酬待遇	16000±

• 教育背景

教育时间	2011 年 7 月 - 2015 年 7 月
学校名称	沈阳工业大学

• 自我评价

1. 积极自我更新，对技术有很高兴趣
2. 逻辑思维清晰，易于沟通相处

• 综合技能

- 熟练掌握 Java 技术栈 Vue.js + npm + Springboot + Gradle + DevOps(sh python),以及 Java 基本技术点(Socket, NIO, AIO, Current, Thread)
- 掌握 Python3 技术(lambda, pip3, yield, 分布式接口,协程等)与自动化部署
- 掌握 UI 开发特性（Vue.js + npm）
- 简历内的技术点可以使用自己的云服务器当场演示

• 项目经验 1

项目名称: PCSD Portfolio & Project Management（软通动力 外派 Lenovo 北京）

项目时间: 2019/03 - 2019/05 (Lenovo 北京)

运行环境: UI: AngularJS + npm Backend: java 8

开发工具: Eclipse(Spring Tool Suite)

技术支持: Backend: Springboot 1.5.4 Gradle 4.8 Selenium-java2.47 TestNG 6.9.10

部署测试: TestNG 6.9.10 Jenkins+Gradle

版本控制: GitLab

项目角色: Backend Developer

项目人数: 8 人

项目描述: 根据用户上传的数据文件，构建 Portfolio 与 Project 计算相关 KPI，主要职责为根据业务需要，构建页面的自动化测试模块，并利用 Jenkins 运行 Gradle 相关的 task，来检验 PPM 项目的可运行性与每次部署后的可靠性

技术描述：通过 Selenium-java，主要根据页面 Element id 与 Selenium-java xpath 属性，将页面的 Element 构建为 Object，解析器其显示的数据与功能是否完整。通过 TestNG 为 Gradle Task 进行分组运行，每一组功能构建一个 Gradle Task

● 项目经验 2

项目名称：Daimler Code Standard and UI design Standard (Alten 欧立腾 外派 Daimler 北京)

项目时间：2018/04 - 2019/02(Daimler 北京)

运行环境：UI: Vue.js + npm Backend: Python3.7 + java 9

开发工具：PyCharm CE IntelliJ IDEA

技术支持：UI: Vue.js Element Intro.js html2canvas.js jquery.js jquery-ui.js
Backend: (Springboot2.0.5+jacoco+checkstyle+Flyway) Junit MyBatis numpy pandas Test

部署测试：Docker18.06 Gradle4.10 Selenium

CI/CD：GoAgent + Pipeline

版本控制：Docker18.06 GitLab

项目角色：从设计文档到持续集成，制定包含整个项目的 DevOps 过程的执行标准

项目人数：10+

项目描述：基础描述：
1. 代码逻辑构建：使用 SpringBoot+Gradle 构建 Http 接口，基本代码符合 domain 与 Clean Code 与 Clean Architecture 的基本要求，TDD 模式保证代码功能与整洁，测试覆盖率要求 98%，本机代码测试要求项目本身绝不出现依赖文件，并要求每个 Http 接口都要有相应的契约测试并形成 Swagger 说明
2. 测试阶段：使用 Python 脚本对后端接口进行集成测试，shell 进行冒烟测试，两步测试与首个 Pipeline commit 都展现在 Agent 上。测试时后端为 docker 容器运行，每次测试要将测试的 docker images 上传到公有 GitLab 上，使用 Selenium 进行页面功能性测试，Jest 进行 js 代码单元测试，要求覆盖率 90%
3. 部署阶段：使用 Pipeline 与 Docker 进行集成部署，必须符合持续交付要求，打包一次部署多次，并且多环境间绝对不允许有交叉

技术描述：
1. Vue.js + npm 进行前端页面开发，测试部署使用 Dockerfile 打包成 Docker images 进行本地测试，
2. Jest 保证测试覆盖率，TDD 模式进行代码构建，每次构建后打包成 Docker 镜像进行版本控制
3. SpringBoot+Gradle 进行接口开发，Gradle 配置 Flyway 与 checkstyle 插件的 task，每次运行 gradle build 后，初始化本地测试使用的 H2 database 与检查代码风格是否符合自定义要求
4. 使用 Selenium 进行页面驱动测试，保证操作的功能可用
5. Python 脚本配置 Flask 进行接口的集成测试，保证 UI 与 backend 代码的交互过程可用，并且配置 pipeline 检查，每次进行环境部署前，先检查接口是否可用
6. 部署前使用自定义 sh 进行冒烟测试，保证当前的进程已经启动
7. 部署时，将前后端的 docker 容器删除，将新打包的前后端 docker images 运行，使用 Nginx docker 进行跨域访问配置，使用 pipeline 配置 Agent 进行 CI/CD 过程

● 项目经验 3

项目名称: CSA (Custodian Saler Approval) 管理销售批准 SPM (Sales Project Manage) 销售项目管理 (大连阿尔泰克 外派 大连 IBM)

项目时间: 2017/02 - 2018/04 (IBM 中国大连分公司)

运行环境: webSphere

开发工具: Eclipse

技术支持: 后端: Jersey + Spring + MyBatis + Springboot + Springbatch + EJB + IBM cloudant (JS) + solr + Jekinese
前端: JavaScript + HTML + Node.js + Ajax

项目角色: Java ETL 层开发与维护, RestAPI 开发, EJB 数据映射维护

项目人数: 63 人

项目描述:

1. CSA 整体功能简介 (用户 30k±):

根据 IBM 瑞士苏黎世分公司 RTC 服务器的产品, 客户数据与 IBM 销售中心供应商提供的客户数据, 将所有的产品数据与客户 (包括已经发生交易的客户与未发生交易但有可能与 IBM 交易的客户) 进行关联度计算, 将最有可能与 IBM 发生交易的客户集中展示给 IBM 交易部门的销售人员, 使销售具有较大的针对性

2. SPM 整体功能简介 (用户 200±):

由 IBM 美国加州圣地亚哥分公司销售经理, 建立销售剧本数据 (本产品针对于哪一国家, 哪一行业, 哪一种业务, 销售人员怎么与客户进行协调沟通的产品资料, 以及客户的基本资料与交易记录等) 帮助销售人员进行对客户的产品推荐, SPM 为销售剧本数据的构建提供基础数据, 并将销售剧本数据与 CSA 的产品客户关联度数据进行关联计算

我组 (共 6 人) 负责 SIOICI (Sales Into Operation Into Client Into) 模块, 即销售剧本数据, 产品交易行为数据, 客户数据的具体信息构建

3. 工作业务流程及职责

- (1) TL 1 人负责与 IBM 巴西圣保罗分公司的用户模块经理进行业务确认与进度跟进汇报, 并对 Team 提供技术支持
- (2) UI 1 人 负责页面开发构建
- (3) Java API 2 人, 主要负责基础 API 构建与 solr 索引构建以及 Jekinese 定时发布任务, 保持模块稳定与 WAS 代理维护, JNDI 数据源维护与构建分布式事务, 部分 EJB 数据映射管理, 发布版本管理
- (4) Java ETL 1 人 (本人), 主要负责 RTC 数据源, 第三方数据源的数据对本地环境的定时更新, 部分 DB2 表的索引, 存储过程维护与优化, 与 API 所需数据的关联度计算业务
- (5) QA 1 人, 负责模块测试

技术描述: (本组项目采取 Webservice 开发):

1. UI 通过异步对 WAS 发出 http 请求, WAS 项目将 UI 所需数据转为 json 格式通过 http 发出
2. Jersey 搭建 RestAPI 供 UI 异步调用, 在 dev 环境进行开发, uat 环境进行测试, prod 上进行实际线上服务
3. 通过 Spring cacheBeanPostProcessor 属性优先缓存部分常用数据到内存, 提高效率
4. 通过 bitmap 构建内存压缩算法, EWAHCompressedBitmap 实现, 将大量常用数据以数字下标形式, 单独存储在 1bit 上, 并且通过常规 map 进行数据关联, 减小内存占用率
5. 通过 Springboot 微服务架构, 对 API 进行解耦, 使发布更轻量, 当需要发布部分代码时不影响整体模块运行

6. 使用 Springboot 构建服务关联多个 Springbatch 的 job, 通过 http 进行 ETL Job 启停操作与消息通知
7. 主页面数据通过定时的 ETL Job, 针对每一个用户的不同情况, 预计算出不同的数据, 相当于为每一个系统登录者构建数据缓存, 使用 IBM cloudant 进行数据迭代
8. 项目后期考虑设计统一的 API 网关, 减小单个 WAS 的运行压力, 并且避免 Server WAS 单个端口崩溃后, 模块运行不了的情况