

# 实习总结答辩



答辩人：谭旭洋 50041968

导师：刘有志 00612965

项目组：MM1.0 APaaS

主管：陈云 00527510



# 目录

---

- 1** 个人简介
- 2** 实习概况
- 3** 学习与产出
- 4** 总结收获

# 个人简介

谭旭洋 中南大学 计算机学院 研二

- 2020.10 蓝桥杯大赛单片机组全国三等奖
- 2021.8 大学生服务外包创新大赛全国二等奖
- 2021.9 “华为杯”大学生物联网设计竞赛全国二等奖
- 2021.10 电子设计竞赛湖南省二等奖
- 2019-2021 多次获三好学生、优秀团干等荣誉
- 2023.10 “华为杯”研究生数学建模竞赛全国二等奖
- 2024.7 华为技术有限公司实习
- 发表一区论文一篇，还有一篇二审中
- 软件著作权三项

实习部门 软件云服务开发部

导师：刘有志 (00612965)

主管：陈云 (00527510)

项目组: MM1.0 APaaS

# 目录

---

1

个人简介

2

实习概况

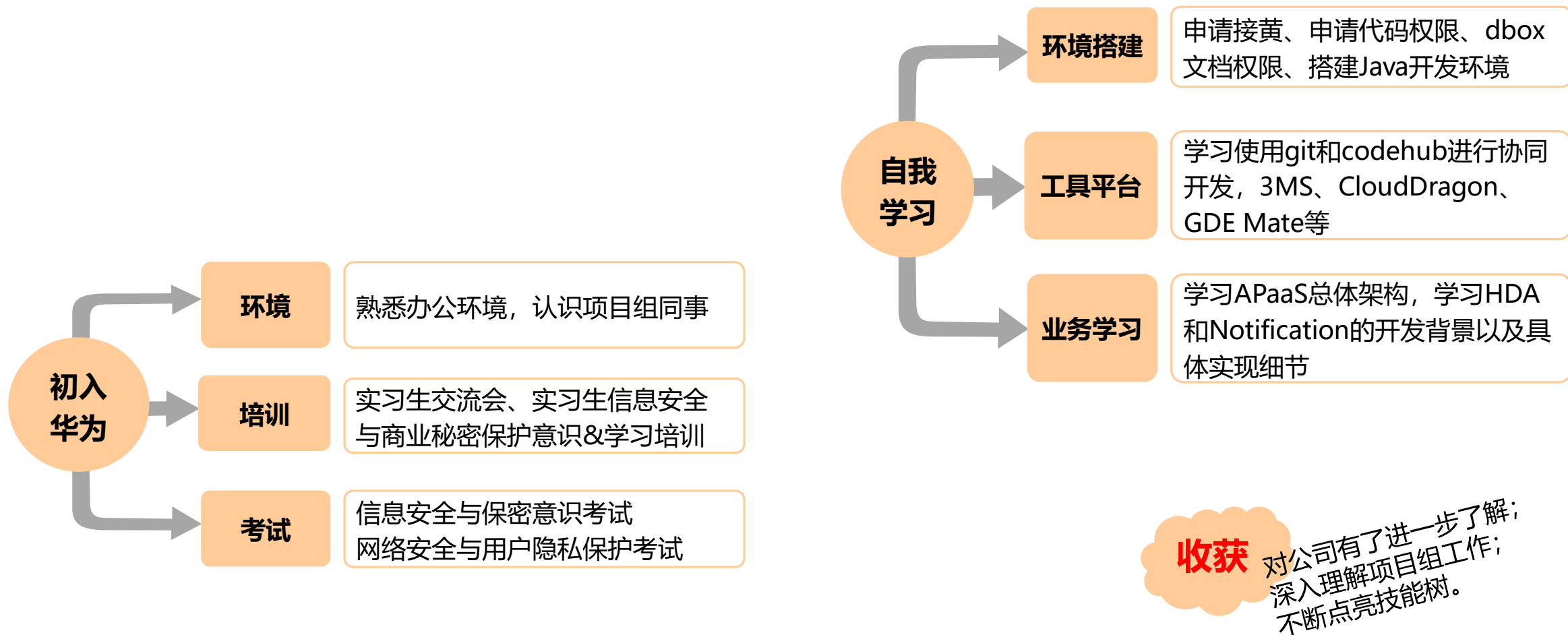
3

学习与产出

4

总结收获

# 实习概况



# 实习概况

时间	学习工作内容	收获	输出
7.3-7.12	1.搭建开发Java环境 2.申请接黄、申请代码权限 3.学习华为Java编码规范 4.熟悉HDA代码功能和结构 5.学习k8s基本命令操作 6.学习APaaS的总体架构	1.学习了华为Java编程规范 2.了解了HDA项目功能结构 3.学习了k8s基本操作	1.输出学习文档 3 篇
7.15-7.26	1.完成HDA接口开发需求 2.学习Apollo配置查询和修改 3.远程debug学习HDA归档代码 4.开发远程debug快速配置工具 5.Notification模块功能和结构学习	1.熟悉了需求评估、开发到测试的整个流程 2.熟悉使用git进行多人协同开发 3.学习了在k8s环境下进行远程debug调试 4.在HDA和Notification中学习了观察者模式和模板方法模式的实际应用	1.输出学习文档 7 篇 2.贡献代码 1600+ 3.发现问题并开发远程debug配置工具，并在小组内推广使用
7.29-8.16	1.HDA接口三个问题单修改 2.DV性能指标采集学习 3.学习十种设计模式 4.帮助测试排查HDA服务启动失败问题 5.根据使用反馈升级远程debug配置工具 6.可信合规提高HDA和NC的UT覆盖率达标	1.接口需要注意入参的校验 2.学习了十种行为型设计模式 3.学会根据启动日志排查容器问题 4.学习了单元测试中常用Mock方法 5.学习了正则表达式匹配的基本规则	1.贡献代码 2800+ 2.输出学习文档 10 篇
8.18 至今	1. 远程debug配置工具组内培训 2. DV智能运维故障检测分析包格式检查 3.远程debug配置工具代码提交到组内仓库 4.答辩准备	1.学习了DV平台的使用操作 2.实习总结	1.贡献代码 700+ 2.远程debug工具开发指南 3.答辩胶片

# 目录

---

1

个人简介

2

实习概况

3

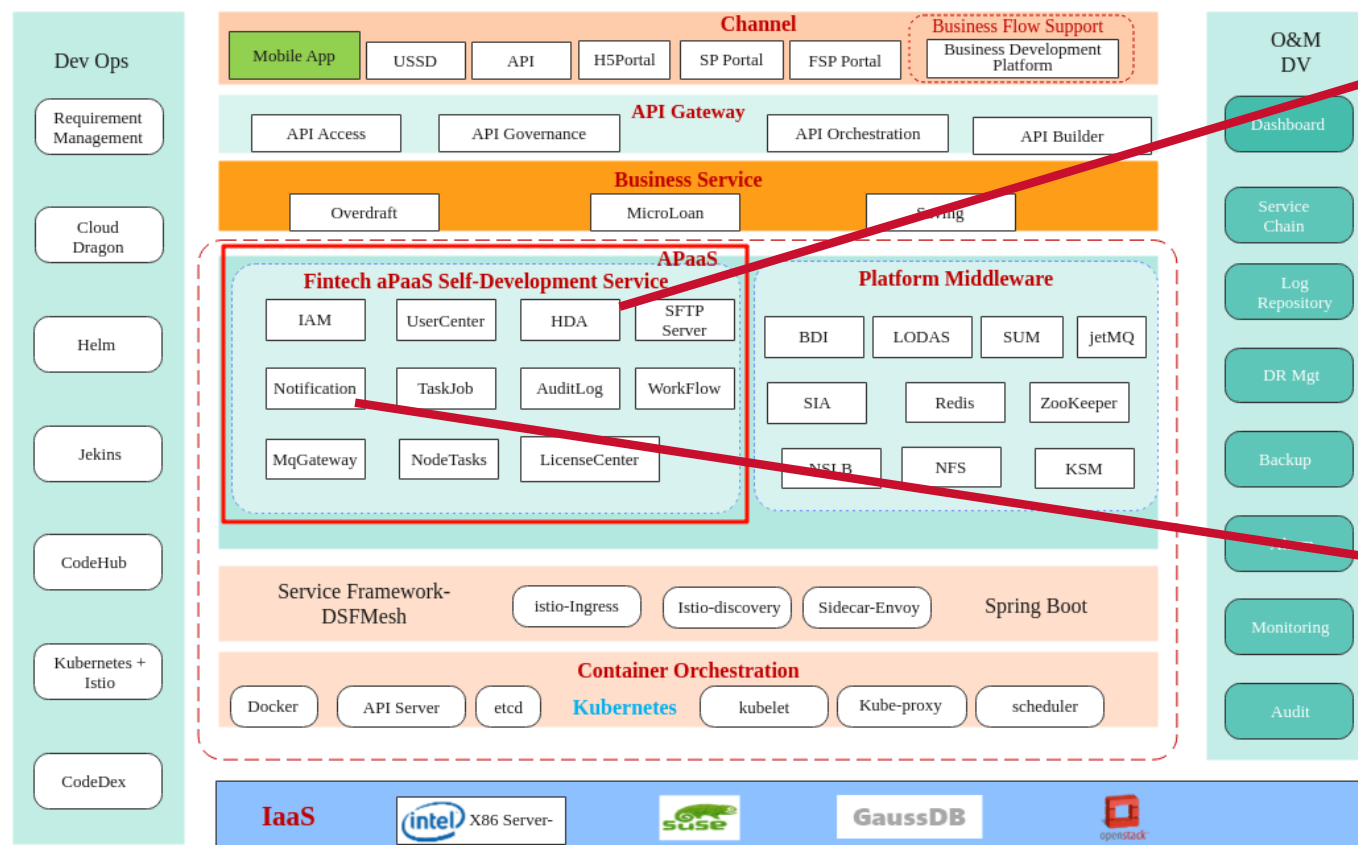
学习与产出

4

总结收获

# 学习与产出-APaaS架构学习

Digital Finance是一种鼓励更多使用移动货币并巩固消费者无现金行为的产品。



## HDA (Historical Data Archive)

背景

随着业务系统的运行，伴随的业务数据也会日积月累的越来越多，如果不及时清理，会降低系统数据的读写速度，影响系统的性能。

功能

通过归档数据，减少系统在线数据量，增强系统在线性能，同时可以对数据起到长期保存的目的

## Notification

背景

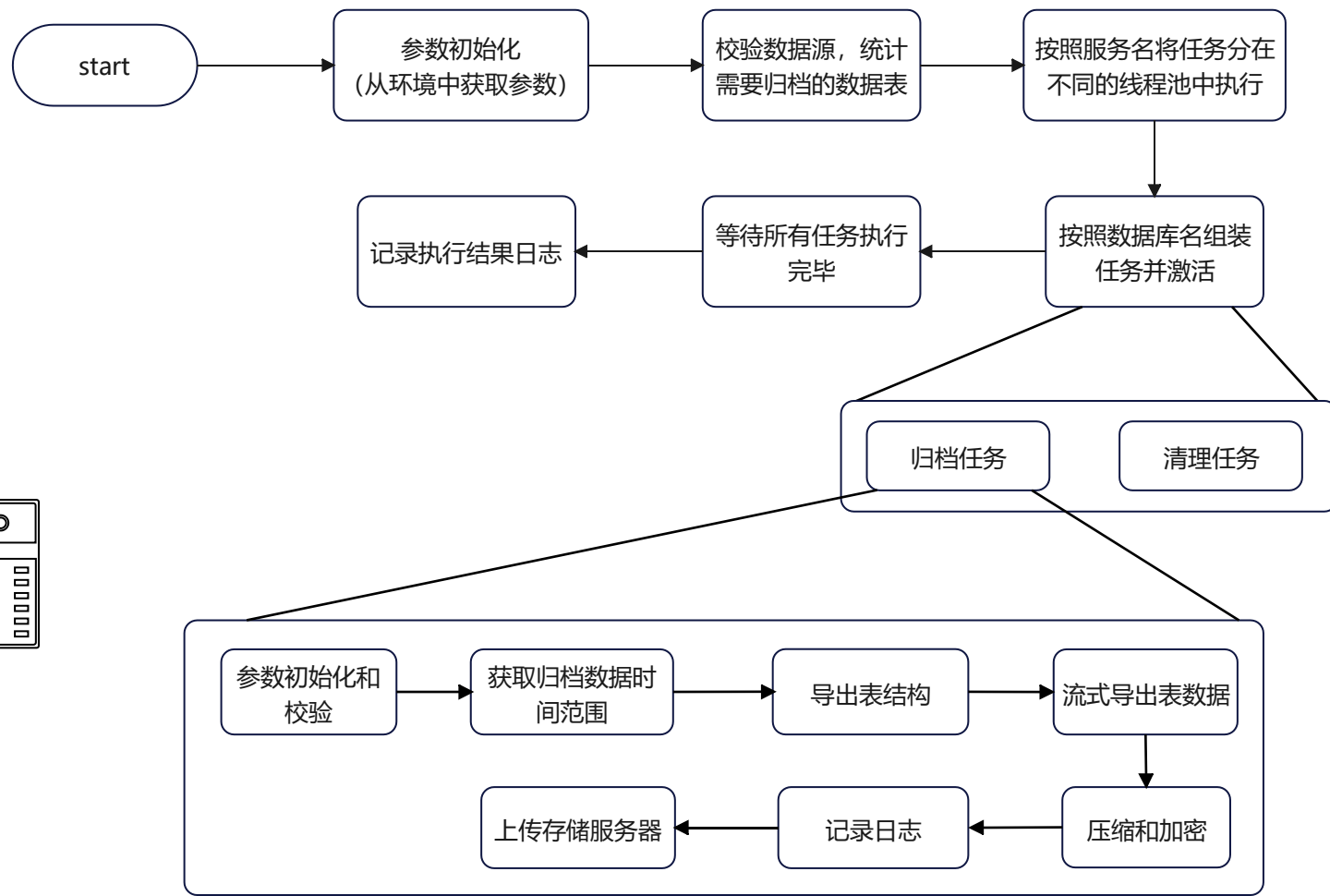
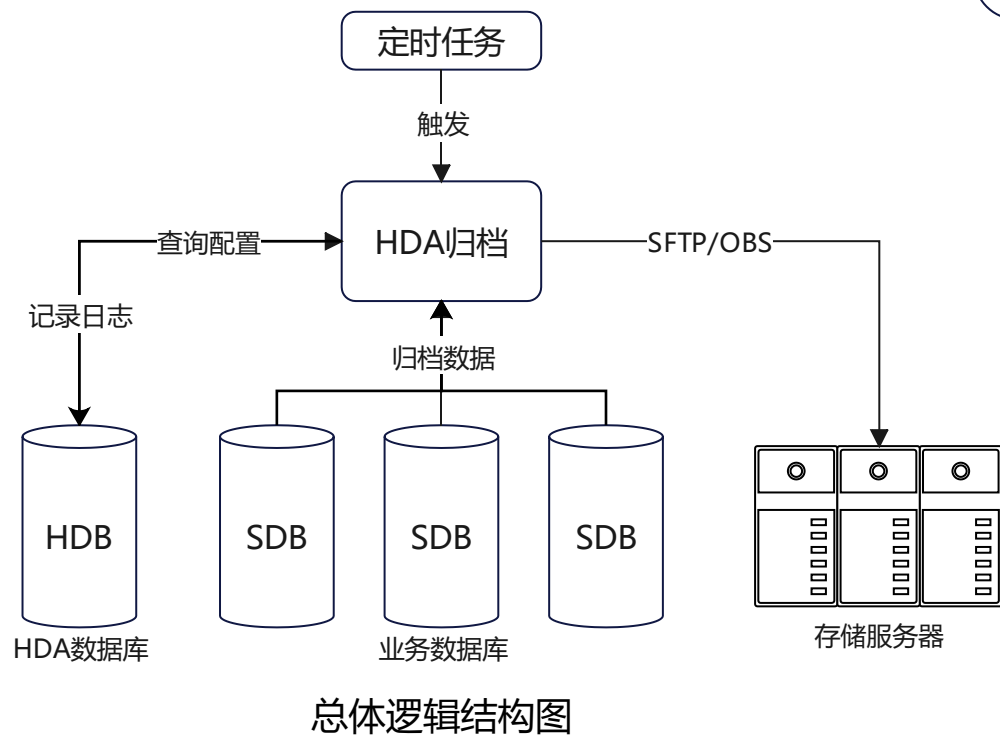
系统在运行时需要通知相关人员处理结果、费用、余额等信息，APaaS需要提供一个通知框架，提供通知相关的配置和发送能力。

功能

- 提供通知时间、通知策略、通知模板的配置
- 支持发送SMS和Email通知
- 支持手动发送通知



# 学习与产出-HDA归档流程



## 需求背景

提供HDA配置表的增删改和分页条件查询接口、HDA日志表的分页条件查询接口

## 流程分析

1. Controller层使用SpringFramework中的@Validated校验入参
2. Service层使用Foundation中提供的PageCondition封装查询条件，进行分页查询
3. Mapper层使用MyBatis的动态SQL-if再次过滤空参数
4. 封装响应类继承自Foundation中的Response，对返回数据进行封装

## 工作

1. 实现HDA\_MAIN\_CFG表的增删改以及分页条件查询接口
2. 实现HDA\_TABLE\_LOG表的分页条件查询接口
3. 转测后测试提的三个问题单修改

## 收获

1. 开发接口时注意对入参进行校验
2. 熟悉了Foundation中的PageCondition的使用
3. 了解需求开发的整个流程

API总数: 5

### hda logs query

POST /log/query hda logs query

### hda config query

POST /config/query hda config query

### create config

POST /config/create 创建HDA归档配置

### delete config

POST /config/delete 删除HDA归档配置

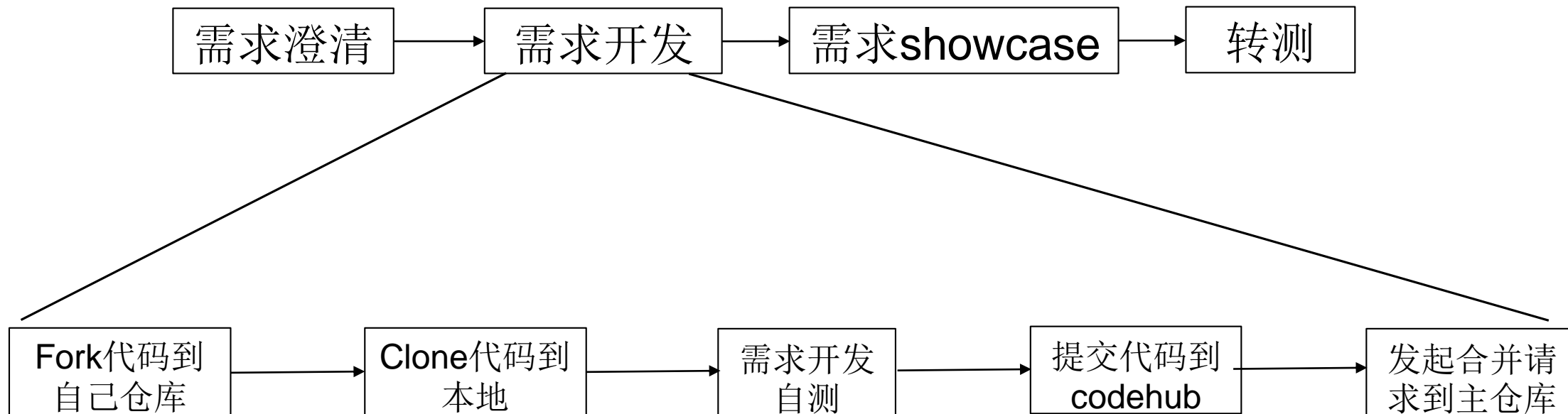
### update config

PUT /config 更新HDA归档配置

需求  
背景

提供HDA配置表的增删改和分页条件查询接口、HDA日志表的分页条件查询接口

## 开发流程



HDA模块

新增用例7  
覆盖率提高6%  
新增代码300+

修改前

修改后

成功率	详情>>
用例总个数	190
成功率	100.000%
成功个数	190
失败个数	0
错误个数	-
跳过个数	0

全量覆盖率	详情>>
行覆盖率 (覆盖...	57.26% 1554 / 2714
类覆盖率 (覆盖...	84.27% 75 / 89
方法覆盖率 (覆...	68.91% 574 / 833
分支覆盖率 (覆...	27.47% 153 / 557

成功率	详情>>
用例总个数	197
成功率	100.000%
成功个数	197
失败个数	0
错误个数	-
跳过个数	0

全量覆盖率	详情>>
行覆盖率 (覆盖...	63.74% 1730 / 2714
类覆盖率 (覆盖...	85.39% 76 / 89
方法覆盖率 (覆...	73.23% 610 / 833
分支覆盖率 (覆...	32.68% 182 / 557

Notification模块

新增用例**130+**  
覆盖率提高**5%**  
新增代码**2500+**

修改前

修改后

成功率	详情>>
用例总个数	610
成功率	100.000%
成功个数	610
失败个数	0
错误个数	-
跳过个数	0

全量覆盖率	详情>>
行覆盖率 (覆盖...)	55.97% 8284 / 14802
类覆盖率 (覆盖...)	85.09% 348 / 409
方法覆盖率 (覆盖...)	63.62% 2377 / 3736
分支覆盖率 (覆盖...)	35.08% 1438 / 4099

成功率	详情>>
用例总个数	742
成功率	100.000%
成功个数	742
失败个数	0
错误个数	-
跳过个数	0

全量覆盖率	详情>>
行覆盖率 (覆盖...)	61.41% 9088 / 14798
类覆盖率 (覆盖...)	87.78% 359 / 409
方法覆盖率 (覆盖...)	68.23% 2549 / 3736
分支覆盖率 (覆盖...)	41.23% 1690 / 4099

# 学习与产出-可信合规UT覆盖率

```
/**
 * 通过反射测试一般的bean对象
 *
 * @author t50041968
 * @since 2024/8/12 9:58
 */
16 inheritors  ± tanxuyang t50041968
public abstract class AbstractBeanTest<T> {

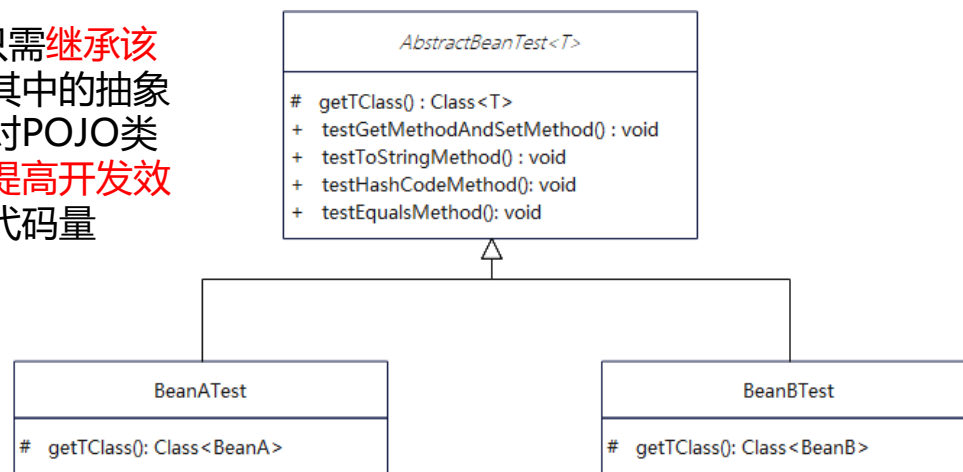
    1 usage
    private static final Log LOGGER = LogFactory.getLog(AbstractBeanTest.class);

    /**
     * getT 获取待测类型
     *
     * @return T
     */
    16 implementations  ± tanxuyang t50041968
    protected abstract Class<T> getTClass();

    ± tanxuyang t50041968
    @Test
    public void testGetMethodAndSetMethod() {
        Class<T> modelClass = getTClass();
        Field[] fields = modelClass.getDeclaredFields();
        for (Field field : fields) {
            // 跳过final方法
            if (Modifier.isFinal(field.getModifiers())) {
                continue;
            }
            String propertyName = field.getName();
            if (field.getGenericType().toString().equals("boolean") && propertyName.startsWith("is")) {
                propertyName = propertyName.substring(beginIndex: 2);
            }
            try {
                PropertyDescriptor pd = new PropertyDescriptor(propertyName, modelClass);
                Method get = pd.getReadMethod();
                Method set = pd.getWriteMethod();
                set.invoke(modelClass.newInstance(), get.invoke(modelClass.newInstance()));
            }
        }
    }
}
```

基于反射机制封装抽象测试类，实现对普通bean对象get、set、equals、toString、hashCode方法的覆盖，覆盖率基本可达100%

POJO测试类只需继承该抽象类，重写其中的抽象方法即可完成对POJO类的测试，极大提高开发效率，降低重复代码量



```
/**
 * SMSMessage 测试类
 *
 * @author t50041968
 * @since 2024/8/12 10:04
 */
± tanxuyang t50041968
public class SMSMessageTest extends AbstractBeanTest<SMSMessage> {

    ± tanxuyang t50041968
    @Override
    protected Class<SMSMessage> getTClass() { return SMSMessage.class; }
}
```

# 学习与产出-远程debug快速配置工具

## 发现问题

微服务架构项目无法在本地启动，调试代码只能打包成镜像到线上环境远程调试，k8s环境下需要修改配置文件添加JVM参数，并暴露出debug端口，每次都是这几步机械化的操作。

## 解决方案

- 1.基于Python使用pyinstaller打包的一个远程debug配置工具
- 2.工具支持一键打包本地代码，部署到线上环境供我们测试
- 3.工具可以自动配置远程debug相关设置，开发人员无需再关心配置过程
- 4.工具支持多历史版本的配置文件备份，可随时进行回退

## 成果

- 1.解放双手，提高开发效率，可至少节省50%的配置修改时间
- 2.在小组内推广使用，获得一致好评
- 3.提交一条自发改进建议，并完成闭环

## 更新日志

- v1.2.2 2024年8月6日
  - ✓ service和deploy名称不同程序无法正常运行
  - ✓ 正则匹配优化
- v1.2.1 2024年7月31日
  - ✓ yaml文件中不存在键闪退问题
  - ✓ debug port校验
- V1.2 2024年7月27日
  - ✓ 增加部署成功检测，容器运行起来之后程序再退出
- V1.1 2024年7月26日
  - bug修复:
    - ✓ win10 cmd控制台输出乱码修复
    - ✓ win10按任意键退出不暂停
  - 新特性:
    - ✓ namespace和服务name输入支持正则匹配
    - ✓ 多次重复执行校验问题
- V1.0 2024年7月23日

组内推广使用

基于使用反馈  
进行迭代

# 学习与产出-小结

## 远程调试学习记录

首先导出deploy和service的配置文件 1kubectl get deploy hda -n t100mpaas -o yaml hda-deploy.yaml2kubectl get service hda -n t100mpaas -o yaml hda-service.yaml 先分别备份一下service文件和deploy文件，cp hda-

2024-07-19 12 0 0 0 0



## 单元测试学习总结

代码 首先可以定义一个BaseTest类，继承自抽象类AbstractTest，AbstractTest中已经定了一套基本的测试框架，部分源码如下： 1@ExtendWith(MockitoExtension.class)2public abstract class AbstractTest {3 4 protected

2024-07-19 8 0 0 0 0



## vim常用快捷键

指令模式 快捷键 功能 q 一般退出 :q! 强制退出不保存 :wq 保存退出 :w 文件名 另存为文件名fileName :set nu 显示行号 :set nonu 取消行号显示 删除 快捷键 功能 dw 删除一个词 dgg 删除至文件开头 dG 删除至文件末尾 dd 删除当前行 d\$ 删除本行光标后面的内容 d0 删除

2024-07-19 7 0 0 0 0

## 模板方法模式

在看HDA代码的时候发现其中对于归档任务和清理任务的设计用了模板方法的设计模式，详细学习记录了基本介绍 模板方法定义了一个算法的步骤，并允许子类为其中的一些步骤提供具体的实现方式。这种模式

2024-07-19 5 0 0 0 0

## 观察者模式学

概述 观察者模式 到通知并自动更 2024-07-19

命令模式 是什么 命令模式允许将请求封装成一个对象（命令对象，包含执行操作所需的所有信息），并将命令

## IDEA注释模板

类注释 1\*\* 2 \* 3 3\* 4\$params\$5\$ 2024-07-10

策略模式 是什么 策略模式是一种行为型设计模式，他定义了一系列算法来使用不同的策略来完成同一件工作，并

## Mobile-Money

权限申请 接黄诗 版本的X64的安 2024-07-05

备忘录模式 基本概念 备忘录模式允许在不暴露对象实现的情况下捕获对象的内部状态，并在对象之外保存这个状态，以后

## 单例模式

是什么 单例模式是一个典型的创建型设计模式，核心思想是保证一个类只存在一个实例，并提供一个全局访问点来访问这个实例。只有

2024-08-05 3 0 0 0 0

## 工厂方法模式

简单工厂模式 简单工厂模式的核心思想是将对象的创建封装在一个工厂类中，吧创建对象的的流程集中在这个工厂类里面。简单工厂包括三个主要角色，工厂类、抽象对象、具体对象。抽象对象描述对象的通用行为，通常

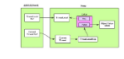
2024-08-05 2 0 0 0 0



## ThreadLocal学习总结

ThreadLocal简介 ThreadLocal顾名思义，就是线程局部变量，也可以叫线程变量。可以让不同的线程访问自己独有的变量，而不会影响其他线程变量，实现了线程隔离级别的变量存储。常见的使用场景包括：会话管理：

2024-08-05 8 0 0 0 0



## 建造者模式

是什么 建造者模式（也被成为生成器模式），是一种创建型设计模式，软件开发过程中有的时候需要创建很复杂



工作内容

贡献代码

修改问题单

完成需求开发

提交改进建议

输出学习文档

产出

5000+

3条

2项

1条

20余篇

<feat><><k8s环境下的远程debug快速配置工具v1.2.2> t50041968 / aPaaSDevops | 1288 | 毛向军 30012995 合并于 2024-08-22 09:21:42 | master 至 master | +714 -0

<test><AR20240710649190><可信合规-UT补充完善> t50041968 / Notification | 1945 | 谢宁杰 00608280 合并于 2024-08-19 14:02:16 | master 至 master | +2583 -97

CI门禁通过 代码检查通过

<test><AR20240710649190><可信合规-DT补充和完善> t50041968 / HDA | 1311 | 谢宁杰 00608280 合并于 2024-08-19 14:04:42 | master 至 master | +332 -53

CI门禁通过 代码检查通过

<test><AR20240710649190><DT补充和完善> t50041968 / HDA | 1310 | 谭旭洋 50041968 关闭于 2024-08-09 09:44:25 | master 至 master | +180 -22

CI门禁失败 代码检查失败

+7 -0

CI门禁通过 代码检查通过

+3 1 / 1 / 1

CI门禁通过 代码检查通过

并且度为0的结点在同一层个节点的二叉树。完全二

CI门禁通过 代码检查通过

对象，而不是从头开始创建。 在原型模式中，通常态来创建，而无需知道具体的创建细节。为什么如

CI门禁通过 代码检查通过

几个配置文件，有的时候还

CI门禁通过 代码检查通过

HDA模块学习总结 开发背景 随着业务系统的运行，伴随的业务数据也会日积月累的越来越多，占用的数据库空间也会越来越大，如果

2024-07-19 8 0 0 0 0

远程调试学习记录 首先导出deploy和service的配置文件 1kubectl get deploy hda -n t100mpaas -o yaml hda-deploy.yaml2kubectl get service hda -n t100mpaas -o yaml hda-service.yaml 先分别备份一下service文件和deploy文件，cp hda-

2024-07-19 12 0 0 0 0

单元测试学习总结 代码 首先可以定义一个BaseTest类，继承自抽象类AbstractTest，AbstractTest中已经定了一套基本的测试框架，部分源码如下： 1@ExtendWith(MockitoExtension.class)2public abstract class AbstractTest {3 4 protected

2024-07-19 8 0 0 0 0

CI门禁通过 代码检查通过





# 目录

---

1

个人简介

2

实习概况

3

学习与产出

4

总结收获

# 总结收获

---

- ✓ 了解了公司各项流程规范、权限申请，包括但不限于考勤管理、信息安全、文档权限、代码权限以及需求设计到开发转测的整个流程。
- ✓ 学习了公司内部各种平台和基础设施的使用，包括w3门户、3MS社区、CodeHub、CloudDragon、GDE Mate、O3大模型、CodeMate等。
- ✓ 学习了HDA和Notification的总体业务背景以及其中的具体实现细节，微服务架构下项目的整体逻辑结构，以及微服务环境下的各种基础组件的作用和基本使用。

总结落实

勤于思考

主动高效

# 致谢

# Thank you.

- ✓ 感谢华为公司能够给我这个实习的机会！
- ✓ 感谢导师刘有志、组长谢宁杰、PL陈云对我的指导和帮助！
- ✓ 感谢项目组的同事们，在我遇到问题的时候无私的帮助！
- ✓ 感谢各位评委的现场指导！

把数字世界带入每个人、每个家庭、  
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and  
organization for a fully connected,  
intelligent world.

Copyright©2018 Huawei Technologies Co., Ltd.  
All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

