|  |  |  |  |
| --- | --- | --- | --- |
| **会议纪要** | | | |
| 微服务开发框架V1.0.0版本评审会议 | | | |
| **一、基本信息** | | | |
| 会议名称 | 微服务开发框架V1.0.0版本评审会议 | 主持人 | 傅慧 |
| 会议日期 | 2018-01-31 | 开始时间 | 14:00 |
| 会议地点 | S01楼1801会议室 | 持续时间 | 4.5小时 |
| 记录人 | 许仲恺 | 审核人 | 傅慧 |
| **二、参加人员**（列出参加会议的人员及所属角色） | | | |
| 傅慧、宋伟伟、宗永洲、王放、张青全、韩晓辰、孙建鹏、王超、许仲恺 | | | |
| **三、会议目标**（简要说明会议的目标） | | | |
| 针对微服务开发框架V1.0.0将要发布的内容进行评审 | | | |
| **四、涉及材料**（列出会议讨论的所有项目资料） | | | |
| 《微服务开发框架V1.0.0版本提交内容》 | | | |
| 1. **会议内容**（记录发言人的观点、意见和建议） | | | |
| 1. 王超介绍基础架构模块拟提交内容，与会人员提出评审意见    1. 介绍注册中心服务，以及部署和使用说明    2. 介绍网关服务，以及部署和使用说明    3. 介绍API管理服务使用说明 2. 张青全介绍安全认证、日志和公共组件模块拟提交内容，与会人员提出评审意见    1. 介绍身份认证服务端架构和客户端配置方法    2. 介绍BSP表结构    3. 介绍Redis公共方法    4. 介绍日志记录配置方法 3. 王放介绍前端拟提交内容，与会人员提出评审意见    1. 介绍前端组件选型结论    2. 介绍开发框架使用方式    3. 介绍开发示例 | | | |
| **六、会议决议**（记录该会议达成的决议） | | | |
| 评审过程中大家的评审意见及应对方案总结如下：   1. 微服务框架中应提供丰富的容错机制，比如注册中心注册完全相同的服务时就提示错误信息，而不是让研发人员调试时才发现错误。应对方案：该问题目前注册中心可以解决，其他组件在设计时也会充分考虑容错性和研发人员易用性问题。 2. 将Eureka提供的界面整改成我们框架自己的页面，同时集成到首页的菜单中，有权限的系统管理员可以看到这个菜单，同时限制直接通过URL访问注册中心服务。应对方案：很好的提议，在后续版本中按照此思路集成，同时，针对注册中心URL访问的控制，需要在网络层通过网络白名单的方式实现。 3. 框架应支持在各业务服务中能够灵活配置各种属性，即时生效。应对方案：这部分需求已规划为配置中心服务，可以实现各业务服务统一管理配置文件，并自动向推送。该服务需要使用Git作为代码管理工具，需要先协调Git管理服务器。 4. 需要对业务服务如何配置到网关进行补充说明。应对方案：今晚完成配置方法的说明，随1.0.0版本发布。 5. 向业务开发人员提供文档时，仅需提供一种最优，或官方推荐的配置方式即可。很多实现配置方式很多，一一说明会增加理解难度，项目也难以统一管理。应对方案：今晚检查提交的说明，均按照这个思路修改，随1.0.0版本发布。 6. 提供的需要部署的组件，在生产环境如何部署？应对方案：由于大部分组件都集成了Spring Boot，可以以jar的方式启动服务，所以开发时可以使用这种方式部署，简便快捷。到生产环境时，我们会针对每个组件提供jar/war/Docker三种部署方式，根据项目需要选择。 7. 提供eclipse插件和新的微服务模板，在新建工程的时候可以通过向导配置各种属性。应对方案：是个很好的提议，目前框架需要有很多东西配置在业务服务中，新建服务配置起来很复杂，我们会规划一个框架专属的eclipse插件，以简化这些工作。 8. 提供带菜单的系统主页面，以方便业务开发。应对方案：会在下个版本中提供。 9. BSP的权限需要支持到应用和按钮。应对方案：可以支持，下个版本提供表结构，具体实现纳入到BSP开发计划中。 10. BSP要支持组织多组织视角，解决业务中组织视角交叉问题。应对方案：很多业务会用到这个场景，必须支持，纳入到BSP开发计划中。 11. 需实际确认通过统一工程访问缓存快，还是访问数据库快。应对方案：考虑到安全问题，我们规划的是由唯一的工程统一访问缓存，对其他业务服务屏蔽对缓存的访问。但这又面临效率问题，这样设计可能会出现访问缓存还不如访问数据库快的情况。因此我们需要做个测试，以测试数据为支持，选择一个方案。 12. BSP支持用户名登陆，同时也支持指定属性登陆，例如手机号、QQ号等。应对方案：可以支持，纳入到BSP开发计划中。 13. 日志链是否可以通过重写，不再让业务开发人员关注日志链的UUID。应对方案：是个好主意，可以让业务开发人员不必特意关注记录日志的细节。计划通过重写log4j的debug/info/warning/error方法和工厂类来实现。同时，计划研究LogBack的日志记录方式，实现自动添加日志链功能。 14. FireFly已经无法与新的开发工具集成，建议更换为Git。应对方案：很有必要，微服务的配置中心组建也基于Git才能使用。因此需要整理一份说明，说清楚想要更换配置管理工具的理由，提交领导，与公司配置管理员协商。 15. 前端页面需要与安全认证服务器进行集成。应对方案：尽快安排集成，在业务开发人员统一培训前出具测试结果。 | | | |