

Checklist方案选型

1.0 需求背景

1.0.1 现状分析

当前场景:项目组在日常工作中需要对任务分解和推动,并通过确认的方式跟踪任务进度.当前缺少对各项工作的进度与内容的整体把控方式,因而使用checklist拆分任务,跟进任务.工作流为创建版本任务->协同拆分专项任务->跟进执行任务

2.0 架构选型

当前系统在选择协同编辑的方式优先考虑了报告平台现有的sharedb协同方式以减少开发运维成本:使用一个常驻的node后端维护协同编辑的内容,持久化时调用使用业务端服务的持久化能力.同时本项目的核心表结构比较轻量,所以考量对比了前后端分离模式和一体化研发模式.

<table><tr><th colspan="2">tb_taskset /* 任务集表 */</th></tr><tr><td>project_id /* 项目id */</td><td>int(11)</td></tr><tr><td>version_name /* 版本号 */</td><td>varchar(63)</td></tr><tr><td>creator /* 创建人 */</td><td>varchar(63)</td></tr><tr><td>name /* 名称 */</td><td>varchar(63)</td></tr><tr><td>manager /* 责任人 */</td><td>varchar(63)</td></tr><tr><td>status /* 状态 */</td><td>int(11)</td></tr><tr><td>is_delete /* 逻辑删 */</td><td>tinyint(4)</td></tr><tr><td>description /* 描述 */</td><td>varchar(511)</td></tr><tr><td>executor /* 执行人 */</td><td>varchar(63)</td></tr><tr><td>create_time /* 创建时间 */</td><td>timestamp</td></tr><tr><td>update_time /* 更新时间 */</td><td>timestamp</td></tr><tr><td>id /* 主键 */</td><td>int(11)</td></tr></table>	tb_taskset /* 任务集表 */		project_id /* 项目id */	int(11)	version_name /* 版本号 */	varchar(63)	creator /* 创建人 */	varchar(63)	name /* 名称 */	varchar(63)	manager /* 责任人 */	varchar(63)	status /* 状态 */	int(11)	is_delete /* 逻辑删 */	tinyint(4)	description /* 描述 */	varchar(511)	executor /* 执行人 */	varchar(63)	create_time /* 创建时间 */	timestamp	update_time /* 更新时间 */	timestamp	id /* 主键 */	int(11)	<table><tr><th colspan="2">tb_template /* 任务模板表 */</th></tr><tr><td>name /* 模板名称 */</td><td>varchar(63)</td></tr><tr><td>description /* 描述 */</td><td>varchar(511)</td></tr><tr><td>creator /* 创建人 */</td><td>varchar(63)</td></tr><tr><td>project_id /* 归属项目id */</td><td>int(11)</td></tr><tr><td>is_public /* 是否公共 */</td><td>tinyint(4)</td></tr><tr><td>manager /* 责任人 */</td><td>varchar(63)</td></tr><tr><td>is_delete /* 逻辑删 */</td><td>tinyint(4)</td></tr><tr><td>create_time /* 创建时间 */</td><td>timestamp</td></tr><tr><td>update_time /* 更新时间 */</td><td>timestamp</td></tr><tr><td>task_data /* 任务集合 */</td><td>varchar(8191)</td></tr><tr><td>id /* 主键 */</td><td>int(11)</td></tr></table>	tb_template /* 任务模板表 */		name /* 模板名称 */	varchar(63)	description /* 描述 */	varchar(511)	creator /* 创建人 */	varchar(63)	project_id /* 归属项目id */	int(11)	is_public /* 是否公共 */	tinyint(4)	manager /* 责任人 */	varchar(63)	is_delete /* 逻辑删 */	tinyint(4)	create_time /* 创建时间 */	timestamp	update_time /* 更新时间 */	timestamp	task_data /* 任务集合 */	varchar(8191)	id /* 主键 */	int(11)
tb_taskset /* 任务集表 */																																																			
project_id /* 项目id */	int(11)																																																		
version_name /* 版本号 */	varchar(63)																																																		
creator /* 创建人 */	varchar(63)																																																		
name /* 名称 */	varchar(63)																																																		
manager /* 责任人 */	varchar(63)																																																		
status /* 状态 */	int(11)																																																		
is_delete /* 逻辑删 */	tinyint(4)																																																		
description /* 描述 */	varchar(511)																																																		
executor /* 执行人 */	varchar(63)																																																		
create_time /* 创建时间 */	timestamp																																																		
update_time /* 更新时间 */	timestamp																																																		
id /* 主键 */	int(11)																																																		
tb_template /* 任务模板表 */																																																			
name /* 模板名称 */	varchar(63)																																																		
description /* 描述 */	varchar(511)																																																		
creator /* 创建人 */	varchar(63)																																																		
project_id /* 归属项目id */	int(11)																																																		
is_public /* 是否公共 */	tinyint(4)																																																		
manager /* 责任人 */	varchar(63)																																																		
is_delete /* 逻辑删 */	tinyint(4)																																																		
create_time /* 创建时间 */	timestamp																																																		
update_time /* 更新时间 */	timestamp																																																		
task_data /* 任务集合 */	varchar(8191)																																																		
id /* 主键 */	int(11)																																																		
<table><tr><th colspan="2">tb_task /* 具体任务 */</th></tr><tr><td>task_content /* 任务内容 */</td><td>varchar(255)</td></tr><tr><td>description /* 描述 */</td><td>varchar(511)</td></tr><tr><td>status /* 任务进行状态 */</td><td>int(11)</td></tr><tr><td>taskset_id /* 父id */</td><td>int(11)</td></tr><tr><td>executor /* 执行人 */</td><td>varchar(63)</td></tr><tr><td>is_delete /* 逻辑删 */</td><td>tinyint(4)</td></tr><tr><td>create_time /* 创建时间 */</td><td>timestamp</td></tr><tr><td>update_time /* 更新时间 */</td><td>timestamp</td></tr><tr><td>id /* 主键 */</td><td>int(11)</td></tr></table>		tb_task /* 具体任务 */		task_content /* 任务内容 */	varchar(255)	description /* 描述 */	varchar(511)	status /* 任务进行状态 */	int(11)	taskset_id /* 父id */	int(11)	executor /* 执行人 */	varchar(63)	is_delete /* 逻辑删 */	tinyint(4)	create_time /* 创建时间 */	timestamp	update_time /* 更新时间 */	timestamp	id /* 主键 */	int(11)																														
tb_task /* 具体任务 */																																																			
task_content /* 任务内容 */	varchar(255)																																																		
description /* 描述 */	varchar(511)																																																		
status /* 任务进行状态 */	int(11)																																																		
taskset_id /* 父id */	int(11)																																																		
executor /* 执行人 */	varchar(63)																																																		
is_delete /* 逻辑删 */	tinyint(4)																																																		
create_time /* 创建时间 */	timestamp																																																		
update_time /* 更新时间 */	timestamp																																																		
id /* 主键 */	int(11)																																																		

核心数据库表:

2.0.1 前后端分离模式

前后端分离模式的优点有:

- 1.分工明确:前端和后端可以独立开发和部署，提高了开发速度。前后端通过API进行交互，使得团队分工明确，各自专注于自己的领域;
- 2.可维护性:代码结构清晰，前端和后端代码库独立，易于维护和升级;
- 3.灵活性与专业性:灵活性更高，允许前端和后端团队根据自己的需求选择最适合的技术栈;

2.0.2 一体化研发

注:此次以阿里巴巴ice.js为例

一体化研发模式的优点有:

- 1.通过一体化的命令行工具进行开发和构建，共享源码和类型定义，降低了沟通成本，提高了开发效率。在这种模式下，前端可以以函数方式请求后端服务，使得开发更贴近现代Web研发的方式;
- 2.零配置:Ice.js的设计理念之一是“零配置”，使得开发者能够快速上手，专注于代码逻辑本身，而不是被构建工具所困扰;
- 3.跨端开发: Ice.js不仅支持网页应用，还支持小程序和Weex等多个平台，实现了跨端开发的一致性，降低了多终端适配的成本和复杂度;

2.0.3分离式 VS 一体式优缺点对比

	专注性	扩展性	跨端开发	学习曲线	兼容性	复杂性
分离式	中	强	弱	无	强	中
一体式	强	弱	强	中	弱	中

前后端分离项目和阿里巴巴Ice.js一体化研发模式各有其优势和劣势。

以本项目体来说,一体化的模式的优点是降低了运维成本和沟通成本,但是学习成本和实战中的踩坑无法预估,业务方面维护成本高,而使用传统前后端分离模式几乎没有学习成本,能快速完成任务,后续在业务上的扩展性也更强.例如使用flask实现业务逻辑,老项目使用的的数据库序列化,参数校验,鉴权以及工具类都可以直接复用,快速实现基本功能需求.

2.0.4分离式 VS 一体式 接入风险

分离式接入风险:

- 1.前后端分离,跨端的网络,数据同步
- 2.多个服务,运维方面成本更高

一体式接入风险:

- 1.学习成本和未知的坑较多,个人角度也不熟悉语言,在项目周期上风险较大
- 2.线上bug的定位难度更高

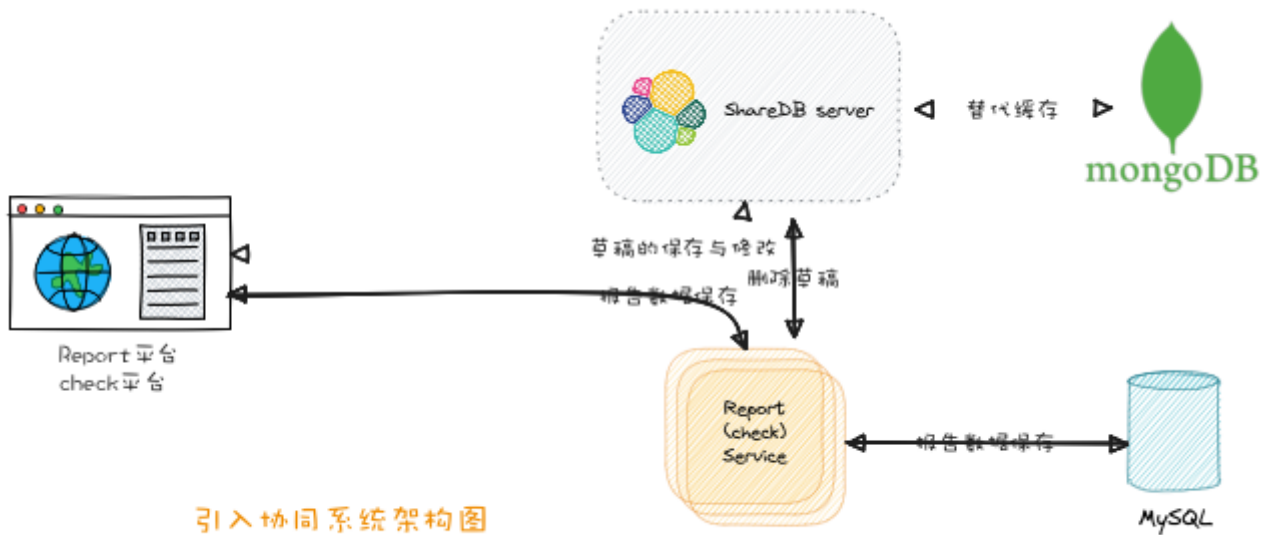
3.0 目标设定

T0:优先完成cheklist本身的业务逻辑,完成任务集模板和任务集的CRUD,对执行任务的进度监控;

T1:完成任务集导入功能,协同编辑任务功能

4.0 技术方案

4.0.1 总系统架构图



4.0.2 技术实施计划与投入预估

后端:

新增checklist业务端服务(预估0d)

2.新增sharedb处理协同部分的数据交互(预估0d)

3.断网关闭网页等异常操作的处理兼容(预估0d)

4.前后端联调

前端:

1.新增checklist业务端服务(预估0d)

2.多组件多collection的连接建立与业务处理(预估0d)

3.断网关闭网页等异常操作的处理兼容(预估0d)

4.前后端联调

4.0.3 风险评估与应对措施

数据隔离:

多组件多collection方案可能导致数据错乱

网络波动以及其他异常处理:

异常状态时的数据保存与恢复

前后端的业务处理一致性

快照版本的恢复策略:

多人操作时,版本快照恢复的严谨性

5.0 效果评估与持续改进

评估维度/指标制定:

评估维度由 开发者 使用方 测试方 三方进行统一制定.

评估指标确定后记录,形成文档保存

持续改进:

测试过程中发现的非主流程阻塞优化改进记录

使用过程中有新的改进型需求进行记录并进行优先级排序

6.0 资源需求与投入计划(根据虚拟小组安排, 相应的链接后续修改)

后端开发(汪李旻):

具体任务拆分详见[redmine][<http://redmine.qa.com/>]

前端开发(翁马轲):

具体任务拆分详见[redmine][<http://redmine.qa.com/>]

测试验收(汪李旻/翁马轲/杨健)

具体任务拆分详见[redmine][<http://redmine.qa.com/>]

相关用例整合详见[用例平台][<http://case.qa.com/home>]

7.0 效果评估与持续改进

推进的check平台进行与用户共创,完善协同工具的使用优化.

8.0 总结

本次checklist平台基于工作场景的任务管理执行而来,选用前后端分离模式开发,后端使用flask以及sharedb搭建框架,核心保障创建版本任务->细分版本任务->执行版本任务->反馈执行进度这一流程,后续可方便扩展阻塞工作通知,工作质量记录等功能,整体优化版本任务的执行流程.

参考链接

[shardb][<https://github.com/share/sharedb>]

[sharedb协同可视化][<https://operational-transformation.github.io/index.html>]

[轻量一体化研发模式][<https://zhuanlan.zhihu.com/p/429417769>]

[开箱即用的ice][<https://developer.aliyun.com/article/807214>]

[ice官网][<https://v3.ice.work/docs/guide/start/>]

[flask][<https://dormousehole.readthedocs.io/en/latest/quickstart.html>]