

Codes 用户手册

目录

Codes 用户手册	1
1、关于 Codes	9
2、Codes 快速指南	9
2.1 功能模型	9
2.2 安装及激活	10
2.3 界面布局	13
2.3.1 简洁布局	14
2.3.1.1 简洁布局	14
2.3.1.2 工作区说明	14
2.3.2 经典布局	15
2.4 全流程项目模板与纯管理项目模板的区别	16
2.5 Codes 敏捷与瀑布的真正融合	16
2.6 Codes 生成式全局看板	18
2.7 Codes 进度管理及填报	18
2.8 Codes 用例执行流程	18
2.9 Codes 缺陷流转或处理缺陷	19
2.10 日报、周报填写	19
2.11 任务流程设置	20
2.12 消息发送	21
2.13 集成配置	21
2.15 项目级的开关设置	22
2.16 重要提示	23
2.17 项目流程	24

2.18. 功能权限及数据权限	25
2.19. 权限、职位菜单及自定义菜单的优先级	25
2.20. 进度及状态	26
2.21. 简洁布局快速入门	26
2.21.1. 管理中心	26
2.21.2.1. 总览	27
2.21.2.2. 仪表盘	27
2.21.2.3. 待办排名:	27
2.21.2.4. 工作负载	28
2.21.2.5. 任务复盘	29
2.21.2.6. 人员产出	29
2.21.2.7. 全局事项	29
2.21.2.8. 管理中心查看迭代计划	30
2.21.2.9. 工时日报、看板、统计分析	30
2.21.2. 阶段 规划或迭代 计划:	31
2.21.2.1. 阶段 规划	31
2.21.2.2. 迭代 计划	32
2.21.3. 工作区:	33
2.21.3.1. 我待办的事项	34
2.21.2.3. 我创建的事项	34
2.21.2.4. 我订阅的事项	35
2.21.2.5. 我待批的工时	35
2.21.4. 项目文档:	35
2.21.5. 工作汇报:	36
2.21.5.1. 日报:	36
2.21.5.2. 个人周报（标准版功能）	37
2.21.5.3. 项目周报（标准版功能）	38
2.22. 需求管理	41
2.22.1. 提示	42
2.22.2. 功能说明	42
2.22.2.1. 需求管理中心主界面	43
2.22.2.2. 创建需求	43
2.22.2.3. 评审及变更	44
2.22.2.4. 需求追踪	44
2.22.2.5. 需求分解为任务	44
2.22.2.6. 需求工时及进度	44
2.23. 用例管理	45
2.23.1. 主界面	45
2.23.2. 产品用例库和公共用例库	46
2.23.3. 线上线下同步及导入	47
2.23.4. 脑图用例	47
2.23.5. 业务场景用例	48
2.24. 执行用例	48

2.24.1. 执行用例过程	48
2.24.2. 执行用例说明	49
2.24.3. 脑图用例执行	50
2.24.4. 用例快速执行	50
2.25. 缺陷跟踪	51
2.25.1. 缺陷管理中心主界面	51
2.25.2. 缺陷流转	52
2.25.3. gitLab 提交代码时和 Codes 缺陷关联	52
2.26. 接口测试流程及接口 Mock	53
2.27. 压测流程	54
2.28. 工具触发执行接口测试	54
2.29. CI CD 流水线编排	54
2.30. 任务管理	55
3、主要功能使用说明	56
3.1 需求管理	56
3.1.1 需求视图及主界面功能	56
3.1.2 需求状态及创建需求	57
3.1.3 需求列表操作说明	57
3.1.4 查看需求拉通的研发活动	58
3.1.5 查看需求变更历史	58
3.1.6 设置负责人	58
3.1.7 单个排期	59
3.1.8 创建子需求及关联需求	60
3.1.9 填写需求进度	60
3.1.10 创建用例，缺陷及任务	60
3.1.11 执行需求下用例	61
3.1.12 填写需求下任务进度	61
3.1.13 评论需求	61
3.1.14 维护需求分类	62
3.1.15 需求分类树切换统计模式	62
3.1.16 自动生成需求任务	63
3.1.17 gitlab 代码提交与需求关联	63
3.2 任务管理	64
3.2.1 概述	64
3.2.2 批量排期	65
3.2.3 Gitlab 代码提交与任务关联	66
3.3 测试流程设置及版本维护	66
3.3.1. 概述	66
3.3.2. 不同流程使用场景	67
3.3.2.1. * 提交问题:	67

3.3.2.2. 测试互验:	67
3.3.2.3. 分析问题:	68
3.3.2.4. 分配问题:	68
3.3.2.5. * 修改问题:	68
3.3.2.6. 开发互检:	68
3.3.2.7. * 分歧仲裁:	68
3.3.2.8. 项目关注:	68
3.1.17.1 * 测试确认:	69
3.4. 缺陷管理	69
3.4.1. 概述	69
3.4.2. 新增缺陷	69
3.4.3. 流转缺陷	70
3.4.4. 流转缺陷入口	72
3.4.5. 缺陷状态集合	72
3.4.6. gitLab 提交代码时和 codes BUG 关联	73
3.5. 测试用例管理	73
3.5.1. 概述	73
3.5.2. 功能主界面	74
3.5.3. 总览	74
3.5.4. 需求树及预分配	75
3.5.4.1. 需求树维护	75
3.5.4.2. 预分配	75
3.5.5. 脑图用例维护	76
3.5.5.1. 基本流程:	76
3.5.5.2. 脑图用例维护功能界面:	77
3.5.5.3. 注意事项:	77
3.5.6. 标准用例维护	78
3.5.7. 标准用例脑图视图	78
3.5.8. 用例导入导出(线上线下同步)	79

3.5.8.1. 概述	79
3.5.8.2. 导入 同步	79
3.5.8.3. 导出 离线处理	81
3.5.8.4. 产品用例库同步用例	81
3.5.9. 用例执行	81
3.5.9.1. 离线执行/处理用例	82
3.5.9.2. 在线执行/处理用例	82
3.5.10. 用例同步	82
3.6. 接口测试	83
3.6.1. 概述:	83
3.6.2. 基本流程:	83
3.6.3. 接口测试主界面:	84
3.6.4. 全局设置	84
3.6.5. BaseUrl,环境变量, 绝对 URL 优先级关系	84
3.6.6. 接口导入	85
3.6.7. 接口加密, 解密, 签名, 前置处理, 后置处理	85
3.6.8. 接口参数化	86
3.6.8.1. 内置参数化	86
3.6.8.2. 内置函数	87
3.6.8.3. 用户自定义采样器	88
3.6.9. 接口间参数传寄	88
3.6.10. 拖拽式接口参数提取	88
3.6.11. 引用提取的参数值	89
3.6.12. 编码接口参数提取	91
3.6.13. 接口断言	91
3.6.14. 拖拽生成断言	91
3.6.15. 编码生成断言	93
3.6.16. 特别提醒	93
3.6.16.1. 复杂键值对转 json	93
3.6.16.2. 美化 json	94
3.6.16.3. 接口前后置 SQL 处理	95
3.6.16.4. URLENCODED Body 设置说明	95
3.6.16.5. 编辑器模式说明	95
3.6.17. 接口依赖拓扑	95
3.6.18. 场景编排	96
3.6.18.1. 概述	96

3.6.18.2. 手动执行接口测试用例	97
3.6.18.3. 数据驱动	97
3.1.17.2 场景中用分配接口模式说明	99
3.6.19. 接口 Mock	100
3.6.20. 接口插件维护	101
3.6.21. 接口定时执行	104
3.6.22. 接口场景环境	104
3.6.23. 接口调用链日志	105
3.6.24. 全局变量使用	105
3.6.25. 接口文档	106
3.6.26. CI/CD 等外具工具触发执行接口测试	106
4、设置	107
4.1 用户管理	107
4.1.1 邀请用户	107
4.1.2 修改用户	108
4.1.3 设置用户是否可查看所有项目	108
4.1.4 密码重置	108
4.1.5 删除用户	108
4.2 用户组管理	109
4.3 权限管理	109
4.3.1 角色管理概述	109
4.3.2 角色人员维护	109
4.3.3 角色权限维护	110
4.4 基础字典维护	110
4.4.1 概述	110
4.4.2 测试用例类	110
4.4.3 缺陷管理类	111

4.4.4 任务管理类	111
4.4.5 用例标签	111
4.5 测试用例库	111
4.5.1 概述	111
4.5.2 公共用例入库	111
4.5.3 产品用例库	112
4.5.4 产品管理	112
4.5.5 入库审核	113
4.6 项目维护	113
4.7 数据源	115
4.8 任务模板	115
4.9 集成配置	116
4.9.1 安全 key	116
4.9.2 企微配置	116
4.9.3 Tapd	117
4.9.4 录制代理	117
4.9.5 邮件配置	118
4.9.6 CI CD 配置	119
5、附录	119
5.1. Codes 通用操作说明	119
5.1.1. 我待处理的缺陷	119
5.1.2. 操作图标	120
5.1.2.1. 人型图标	120
5.1.2.2. 树型图标	120

- 5.1.2.3. 列表切换图标 120
- 5.1.3. 公共用例库与产品用例库的区别 120
- 5.1.4. 业务场景与迭代的关系 121
- 5.1.5. 升级及备份 121
- 5.1.6. gitlab 与 Codes 联动配置 121
- 5.1.7. Codes 与第三方 IM 联动配置 122
 - 5.1.7.1. 与企业微联动 122
 - 5.1.7.2. 与飞书联动 123
 - 5.1.7.3. 与钉钉联动 124

1、关于 Codes

Codes 重新定义 SaaS 模式 = 云端认证 + 程序及数据本地安装 + 不限功能 + 30 人免费

Codes 是一个 高效、简洁、轻量的一站式研发项目管理平台。包含需求管理，任务管理，测试管理，缺陷管理，自动化测试，cicd 等功能； Codes 帮助企业加速融合研发、测试、运维一体化进程。商业版不限功能，本地安装只限用户数，30 个用户免费。

2、Codes 快速指南

2.1 功能模型及敏捷开发流程

2.1.1 功能模型



图 2-1

2.1.2 Codes 敏捷开发流程



产品维护好需求（需求管理或需求池中）---->研发经理排迭代---->迭代中研发人员拆分需求为任务---->测试人员编写测试用例及对需求进行测试---->研发开发完后，设置测试为提交测试状态---->测试人员进行测试并设置迭代为测试状态---->准备发布，在发布中维护发布的所要作的一系事事项。

如还需要更粗粒度的阶段或里程碑，项目的 PM 可经维护这些里程碑，里程碑的进度缺省用其他的迭代进度来推算，也可以设置为手动维护里程碑的进度。

在上述一系列过程中进度管理，Codes 只有通过写工时日报才能填写进度，填写了工时后自动动变更作任务及需求的状态，如不填工时，只能手动更改他们状态。

工时日报详见《[记 Codes 研发管理平台——日报与工时融合集中式填报的创新实现](#)》

2.2 安装及激活

(1) **安装环境**：支持 windows 和 linux 。Windows 安装一键安装包；linux 主要是 Docker 环境，提供了 docker 及 docker compose 的 CURL 一键安装方式，K8s 需要自行配置。

(2) **升级**: windows 有一键安装包, linux 升级和安装一样也是执行 CURL, 只是 URL 上带有版本。

不管那种环境都是 0 配置升级。

(3) **激活**: 安装后如果没有激活, 自动跳到激活页面, 输入在 Codes 官网注册的帐号的租户 ID 便可激活, 激活后, 通过发邀请的方式加入其他用户。切入官网上只需要注册一次。

(4) Linux 下安装后重新安装

停掉服务: compose 方式

```
docker-compose( 或 docker compose ) -f $CODES_HOME/docker-compose.yml
```

down

停掉服务: docker 方式

```
docker stop codes_web_server codes_redis_server codes_mysql_server
```

```
codes_preview_server
```

先把/etc/profile 文件中两个 CODES 打头的变量删了, 然后 source /etc/profile, 然后重开一个 SSH 窗口, 一定重开, 然后 echo \$CODES_HOME 打印出来为空, 就可以把原来的 Codes 安装目录下 codes_config 删掉, 再重装。这是重装哈, 重装原有的数据是没有了的。如是 docker 版还要执行下面命令手动删除这 4 个容器,

```
docker rm codes_web_server codes_redis_server codes_mysql_server
```

```
codes_preview_server
```

升级的话执行官网下载页, 最新的 CURL 脚本, 会自动升级

如发布了新版, 当前您的版本不是最新版, 登录 Codes 时, 右上的头像图标会有红点, 点击查看查更新日志

(5) windows 下安装后重新安装

在控制面板中卸载之前安装的 Codes,然后再执行安装程序

(6) 常见安装问题

1、ubuntu 下 不要用 snap 经常会出现升级后，镜像还是用的老版本

```
root@ubuntu: /codes/codes_config/docker-compose.yml: the following VOLUMES are not found in the image: codes_backend WEB-INF/classes/application.yml
[+] Running 5/6
✓ Network codes_config_codes-net Created 0.1s
✓ Network codes_config_default Created 0.1s
✓ Container codes_preview_server_ Started 1.1s
✓ Container codes_redis_server_ Started 1.1s
✓ Container codes_mysql_server_ Started 1.2s
i Container codes_web_server Starting 1.6s
Error response from daemon: failed to create task for container: failed to create shim task: OCI runtime create failed: runc create failed: unable to start container process: error during container init: error mounting "/codes/codes_config/codes/config/application.yml" to rootfs: cannot create subdirectories in "/var/snap/docker/common/var-lib-docker/overlay2/22f7ab38720851352163c689244ea7275e752d85ca0de3406615a0dfc9016b41/merged/usr/local/tomcat/webapps/codesBackend/WEB-INF/classes/application.yml": not a directory: unknown: Are you trying to mount a directory onto a file (or vice-versa)? Check if the specified host path exists and is the expected type
root@ubuntu:/codes/codes_config#
```

2、Ubuntu , cannot change memory protections 这种错，要增加一个安全配置

```
2024-07-17 20:02:02+08:00 [ERROR] [Entrypoint]: mysqld failed while attempting to check config command was: mysqld --lower-case-table-names=1 --verbose --help --log-bin-index=/tmp/tmp.xo09XGcUeU mysqld: error while loading shared libraries: libpthread.so.0: cannot change memory protections
```

让 appArmor 不要监控 MySQL

解决方案

可以通过以下任何一种方法解决。

从AppArmor监控中排除MySQL

```
root@ubuntu:~# ll /etc/apparmor.d/usr.sbin.mysqld
-rw-r--r-- 1 root root 1520 Dec 10 2020 /etc/apparmor.d/usr.sbin.mysqld
root@ubuntu:~#
```

容器内安装完mysql之后

```
1 | sudo ln -s /etc/apparmor.d/usr.sbin.mysqld /etc/apparmor.d/disable/
2 | sudo apparmor_parser -R /etc/apparmor.d/usr.sbin.mysqld
```

3、redis 不断重启，查看日志提示找不到配置文件，原因是少数机器不认按目录映射

更改 docker-compose 配置如下图所示，把目录映射改为文件映射


```
codes_redis_server:
  image: registry.cn-zhangjiakou.aliyuncs.com/codes_work/redis:7.0
  hostname: redis
  container_name: codes_redis_server
  restart: always
  privileged: true
  volumes:
    - ./redis/data:/data
    - ./redis/logs:/logs
    - ./redis/conf:/etc/redis
    #少数机器不认按目录映射，需要把上面一行注掉（加了#），把下面这行去掉#
    #- ./redis/redis.conf:/etc/redis/redis.conf
  environment:
    TZ: Asia/Shanghai # 设置时区
  networks:
    - codes-net
```

2.3 界面布局

有**简洁布局**和**经典布局**两种布局模式，通过右上的切换开关来切换，缺省为经典布局。

2.3.1 简洁布局

有面向管理人员的(1)管理中心；有面向基层执行人员的(2)工作区，有新订阅的事项生产时会自动推到工作区并显示；还有一个面向 PM 的(3)阶段|规划或迭代计划；最后还有一个就是创建各类事项的（4）全局创建,也就是创建各项事项。

2.3.1.1 简洁布局

如图 2-2 所示：



图 2-2

2.3.1.2 工作区说明

简洁布局缺省首页就是工作区，当然也可以在管理中心中点击右上的图钉图标设置其他页面为首页。工作区缺省显示泳道视图，当然也可切换为表格视图，时间轴视图，泳道的右上角跳转图标可以跳转到相关功能中心，对于执行人员在工作区更改相关事项的状态即可。每个泳道是一个队列，办理后的事项就自动出列（从泳道上消失，同一泳道内其他待办理的事项自动“顶”上来），当然也可以

主动下滑加载更多事项。工作区**我待办的事项**，显示的是指派给我的事项，也就是被动订阅的事件；

工作区**我订阅的事项**，显示我主动订阅的事项。如图 2-3 所示

跳转到对应功能中心，需求管理、任务管理、缺陷管理、用例管理

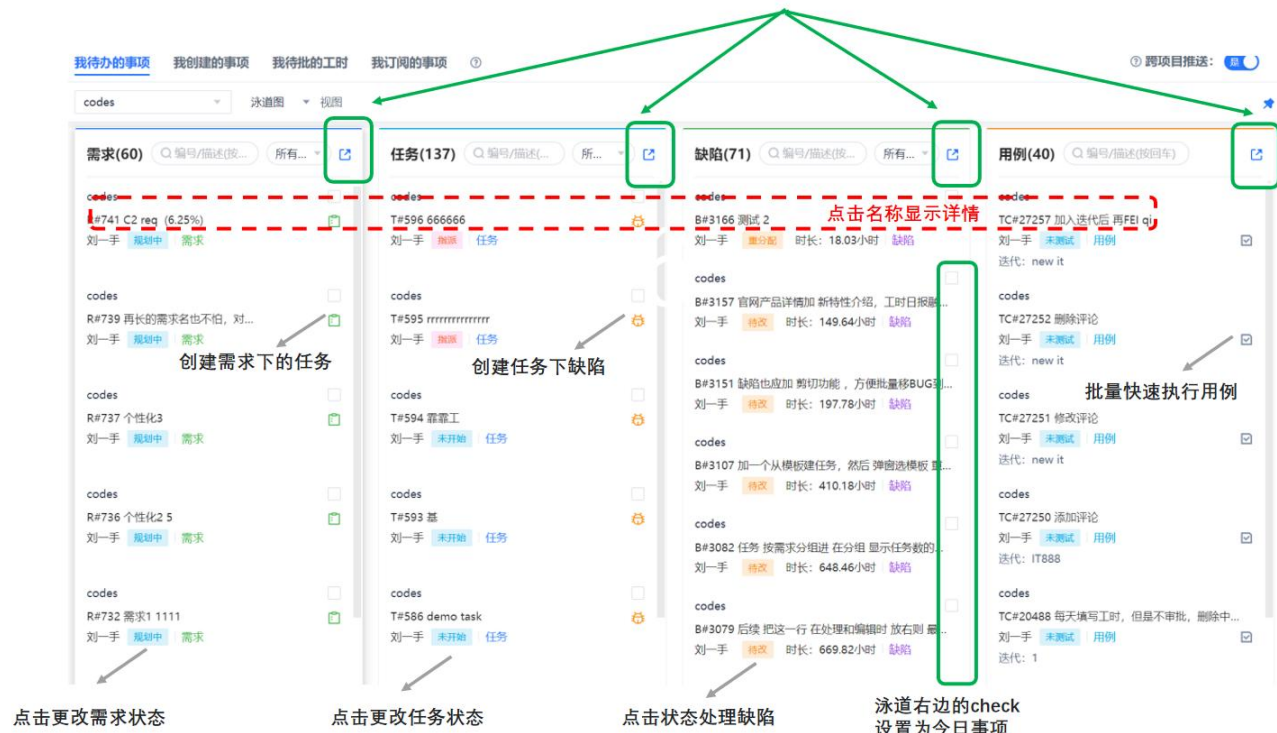


图 2-3

2.3.2 经典布局

经典布局可以自行配置菜单，在权限范围内定制。缺省一个帐户从所属职位继承菜单配置，如所属职位没配置菜单，则按权限来显示菜单。测试相关的功能菜单只有全流程项目时才显示，纯管理项目不显示测试相关的功能菜单。如图 2-4 所示



图 2-4

2.4 全流程项目模板与纯管理项目模板的区别

全流程项目支持从需求到测试，且流程项目的迭代下也有测试相关功能。纯管理项目只有需求管理，任务管理的项目管理功能，无测试相关的功能。这两种项目类型可以随时修改，只是全流程的项目，需要配置缺陷处理流程。

2.5 Codes 敏捷与瀑布的真正融合

Codes 中瀑布和敏捷模式的区别，只是在显示层了主线的不同，底层是一套数据。只是 Codes 中随时可在敏捷视图和瀑布视图间切换，敏捷时以迭代为主线索，瀑布时以阶段为主线索。而不是同类产品融合只是功能菜单上体现的融合，Codes 是数据上的融合。详见公众号上[《研发管理平台——瀑布与敏捷模式融合的创新实现》](#)

只有项目的 PM 才可维护项目的阶段数据，项目阶段也可以从选择的模板中自动生成，生成后再手动维护。

阶段用来作做粗的规化，如里程碑等，迭代用于定义阶段下的具体不同的执行计划。阶段的进度缺省是由阶段下迭代计划的进度自动计算，也可改为手动填写。

一个标准的敏捷的项目过程如下



图 2-5

如果用敏捷模式，但是又需要把宏观的阶段的进展汇报，PM 可以后期维护阶段，再批量把迭代分配到相关阶段下。

一个标准的瀑布与敏捷融合的项目过程如下



图 2-6

2.6 Codes 生成式全局看板

同类产品看板，是物理看板的电子化，使用不便。数据都平台化了，Codes 中平台化的基础上进行创新的看板实现，也就是采用全局生成式看板，在整个系统中只有管理中心有一个全局看板，通过定制不同查询条件来定制不同看板，并可保存；不同的事项，虽然有不同的状态，完全可以泛化为看板上共同的几个泳道所对应的状态。详见《[Codes 研发管理平台——生成式全局看板的创新实现](#)》。

点击任务标题查看任务详情，如延期会显示一个时钟图标；点击缺陷状态，可处理缺陷；点击需求名称，显示需求详情；点击用例状态可执行用例；点击迭代名，可查看迭代报告，点击执行迭代，可进入迭代事项处理界面，点击任务或需求的状态可更改他们的状态。

2.7 Codes 进度管理及填报

任务和需求的状态可以手动更改，但是他们的进度必须填报工时后，才能自动算，不能直接填写进度。填写工时后，任务和需求的状态了会发生变更。工时填报 Codes 中是和日报一起了，集中式填写，也就是说必须写日报才能填工时，且工时不是每个事项单独填写，一个一个的填写非常不方便。详见《[记 Codes 研发管理平台——日报与工时融合集中式填报的创新实现](#)》

2.8 Codes 用例执行流程

Codes 中没有同类工具中的测试计划，而是把测试用例分配，测试用例执行等放到迭代下，在迭代下执行用例。这也是 Codes 敏捷测试的一个特点，以迭代来组织测试。详见《[记 Codes 研发管理平台——多事项闭环迭代的创新实现](#)》

分为四步：先分配用例到迭代，再分配执行人，执行人执行用例，查看执行结果。如图 2-7 所示

如果只用脑图用例，直接把脑图文件分到一个或多个迭代下，然后直接在脑图中执行用例，只是在执行前先切换到在哪个迭代下执行即可，从用例管理中心脑图维护中进入。可以导入 Xmind ，也可直接在 web 编写脑图。



图 2-7

2.9 Codes 缺陷流转或处理缺陷

在所有缺陷显示的页面，都是点击“**缺陷状态**”，来更改其状态并流转到其他人，点击**缺陷描述**，如无修改权限显示的是详情，否则显示的修改页面。具体缺陷处理流程的配置详见《[记 Codes 研发管理平台——流程驱动缺陷管理的创新实现](#)》。

2.10 日报、周报填写

工时日报中，如一个需求太简单没有分解为任务，可以为需求填写工时，如需求下有子需求或是任务，不能填写工时，通过其子需求或任务的工时来推算父需求的进度。个人日报填写后，会自动生成项目日报：

Codes 中工时日报缺省是要审批的可以是 PM 或是部门负责人（不一定和公司行政组织一一对

应)，也可以关掉审批开关，不用审批，不用审批时，日报工时提交后就不能修改。也可从不同层面查看工时统计，按项目，按迭代，按部，按人员，除了按人员外，其他几个统计可以下钻。工时审批后会自动计算相关进度，如需求，任务的进度。只有需求或是任务有了进度数据，才能推算出项目进度，迭代进度。

提示：进度是通过工时来时行计算的，不填写工时，任务和需求的进度就为 0。工时和日报整合一起，方便填写日报及以批量形式填写工时。如填写日报时，关联了风险事项，还能在工作台---》总览---》风险分析中以拓补图的方式查看风险事项。

日报由今日事项、待处理事项、其他事项、以及明日计划组成。日报提交后可以按配置发往第三方平台。详见《[记 Codes 研发管理平台——日报与工时融合集中式填报的创新实现](#)》

日报提交后缺省是需要审批工时，可以关掉开关。审批时可以是 PM 审批也可以是部门负责人，审批时不存在通不通过，主要是可以更改提交的工时。

2.1.0 以上版本还支持个人周报入项目周报



图 2-8

2.11 任务流程设置

只有不关联需求的任务，也就是独立任务才会走流程，关联了需求的任务随需求走流程。独立任务的流程缺省是开启的，可以手动关闭。**开启流程，不关联需求的任务将走审批流程（填写人指派任务----》被指派排期----》负责人审批----》任务实现）**。审批主要是审工期，审的时候可以更改不

能驳回



图 2-9

2.12 消息发送

Codes 有事项指派时，如建了需求、任务、缺陷时指派给了某人，被指派人会收到邮件，钉钉，飞书或企微。如不用发邮件可以在集成配置中邮件设置中关闭发邮件，缺陷相关的发送开在项目的测试流程中配置。发邮件不建议用官方代发，因为用的人多，发邮件太多经常会被拒退回邮件，建议配置自己的 SMTP 发件服务器，公司邮箱，QQ 等都可以。

2.13 集成配置

主要配置一些开关以及和外部系统的交互，如图 2-10 所示。安全 key 主要用于其他第三方系统回调 Codes 时安全认证，比如代码提交关联时 gitlab webhook 中回调 Codes 时 要带着安全 key，CI/CD 也是一样的用法。邮件配置配置 SMTP 发邮件帐号信息。



图 2-10

2.15 项目级的开关设置

在项目维护中进行维护且是项目级的开关。

缺陷修复必须关联代码提交：开上时，在把缺陷改为“已改”状态时，会检查关联了代码提交没有，如没关联侧不能更改为“已改”状态时，这个开关缺省是关掉的。

需求层级：默认为 2 层,可以更改，不建议有太多的需求层级。如设置为 3 层，需求只能最多有三层分级。

用例单位成本：设置用例一成本单位代表多少分钟，缺省为 1 分钟。如用例单位成本设置为 5 分钟，某个用例的执行成本为 2，且执行这个用例记为 $2 * 5 = 10$ 分钟。

阶段 | 里程碑模板：设置引用的模板，随时可以修改，如果选择了 A 模板，同时又手动维护了两个里程碑，这时更改为 B 模板，然后再改为 A 模板，数据为之前 A 模板和之前 A 模板时手动建的。

需求模板：如果需求分类没有指定需求模板，则引用项目这里配置的。需求模板，指需求的格式模板。

项目分类：Codes 中没有项目级的概念，后续用分类来当项目级进行统计分析等。

缺省项目视图：在敏捷和瀑布之间来选择，切换项目后按这里的设置显示缺省的项目视图，当然也可以手动再切换。

需求流转发消息、任务发消息，是发消息的开关,以钉钉还是飞书，还是企微发送，就看集成配置

项目名称 *

研发部门 *

开始日期 *

结束日期 *

项目人员 *

产品人员 *

飞书地址

关联Tapd项目

阶段 | 里程碑模板 无

缺陷项目视图 敏捷

需求流转发消息

任务发消息

需求粒度(天) 5

需求层级

网关上下文

项目经理 *

钉钉地址

用例单位成本 1 分钟

需求模板 无

项目分类 请选择项目分类

缺陷修复必须关联代码提交

确定 重置

图 2-11

2.16 重要提示

表列表头、表单中，弹窗中只要有问号图标，指上去会有提示，是对对应数据的说明。当对一些关键数据有疑问，就可指上去查看提示信息

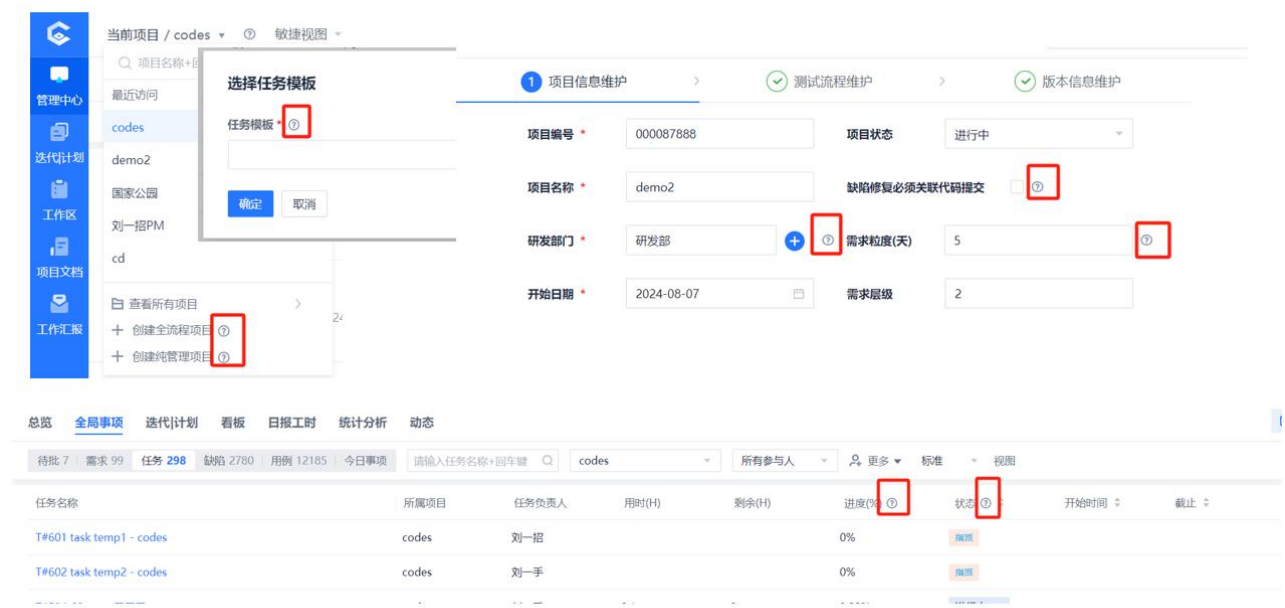


图 2-12

2.17 项目流程

创建项目---->创建阶段（可选）---->创建需求---->创建迭代---->在迭代下分解需求为任务（创建任务）---->编写用例---->实现任务相关功能---->提交工时日报---->CICD(可选流程)---->执行用例---->提交缺陷

提示：在 Codes 中围绕迭代来展开研发工作，通过工时日报，批量式填写工时及工作日报，跟踪项目情况，实现可追溯的日常工作。当把需求分配到迭代时，会自动把需求下已有任务分配到迭代下，在迭代下手动分配任到迭代时，只能选择没关联需求的任务，我们鼓励标准做法，分配需求到迭代下，然后在迭代下把需求分析为任务。

需求管理 >>> **迭代规划** >>> 迭代开发 >>> 缺陷追踪 >>> 迭代进度跟进 >>> 迭代回顾 >>> 发布管理

产品负责人根据需求优先级，将高优先级的需求规划至对应迭代，在迭代计划会议上与团队一起进行迭代评审，确定好当前迭代要完成哪些需求之后，研发人员即可对迭代需求分解，拆分成各类研发任务和预估工时



图 2-13

2.18.功能权限及数据权限

帐户从所属角色获得功能权限，一个帐户所拥有的权限，是由他所属的角色的权限合集组成。

数据权限，普通帐户只能查看自己所参与的项目；管理员能查看所有项目。

功能权限决定了一个由帐户可操作什么功能，数据权限用定义了一个帐户数据的可视范围。

数据权限只用租户可以设置，如图所示，用列表中，有一个查看所有项目的开关，开了就是管理员。

用户管理

组管理

权限管理

姓名/邮箱+回车键

<input type="checkbox"/> 姓名	联系邮箱	联系电话	部门	查看所有项目 ①	状态
<input type="checkbox"/> 罗工	3750050@qq.com	13678913678	产品部	<div><div></div>否</div>	<div>启用</div>
<input type="checkbox"/> qa1	qa1@qq.com	11111111111	测试部	<div><div></div>否</div>	<div>启用</div>
<input type="checkbox"/> xinlaide	qa2@qq.com	12345678901		<div><div></div>否</div>	<div>启用</div>

图 2-14

2.19.权限、职位菜单及自定义菜单的优先级

如切换到经典布局，菜单的显示缺省根据权限来显示，如果定制了使用菜单就按定制的来显示，定制菜单时，如没权限定制时是灰的不可选的状态。为了简化配置，在设置下还可有一个对职位对行的菜单配

置，如果所属的职位配置了某菜单，但是没有相关权限，显示的时候一样不会显示无权限的菜单。也就是说一切都基于拥有的权限，然后才是自定义菜单。

2.20.进度及状态

需求和任务的进度，只有在写日报时填写了工时才能计算出来，不能手动更改；任务状态可手动任意修改，有工时时也会自动更改任务状态；需求的状态如果手动更改，先要变为“实现中”，才能在更改状态的页面看到其他可选的状态，同样提交了工时，需求的状态了会自动变更，如把迭代设置为提交测试，会把迭代下所有需求更改不测试中的状态。

2.21.简洁布局快速入门

以简洁布局为例来说明 5 个功能区：1 管理中心、2 阶段 | 规化或迭代计划、3 工作区、4 项目文档、5 工作汇报。经典布局可查看前面章节中的[界面布局](#)，经典布局时管理中心下的子项可通过菜单定制自行配置。

2.21.1.管理中心

面向 PM 之类的管理人员，主要是一些全局类的事项查看，统计分析以及日常项目级的过程管理的功能。分为总览、全局事项、迭代计划、看板、日报工时、统计分析 6 个分项。管理中心和每个页面右上有一个图钉图标可以设置这个页面为 Codes 首页。

管理中心中还有一个**局部项目**的下拉**切换**，和全局项目切换是联动的，但是当选择所有项目时不联动，对于**管理人员可以切换到的所有项目**，对于**普通人可以换换为我参与的所有项目**。如是管理员有一个缺省显示所有项目的开关，勾上时，切换到管理中心，局部项目缺省就是所有项目。如图所示：



图 2-15

2.21.2.1. 总览

总览下有：仪表盘、待办排名、工作负载、人员产出、任务复盘、风险分析。

2.21.2.2. 仪表盘

当在仪表盘中切换到某个具体项目时，还会显示阶段 | 里程碑信息，如果当前项目项目经理还可维护里程碑信息。如图所示：

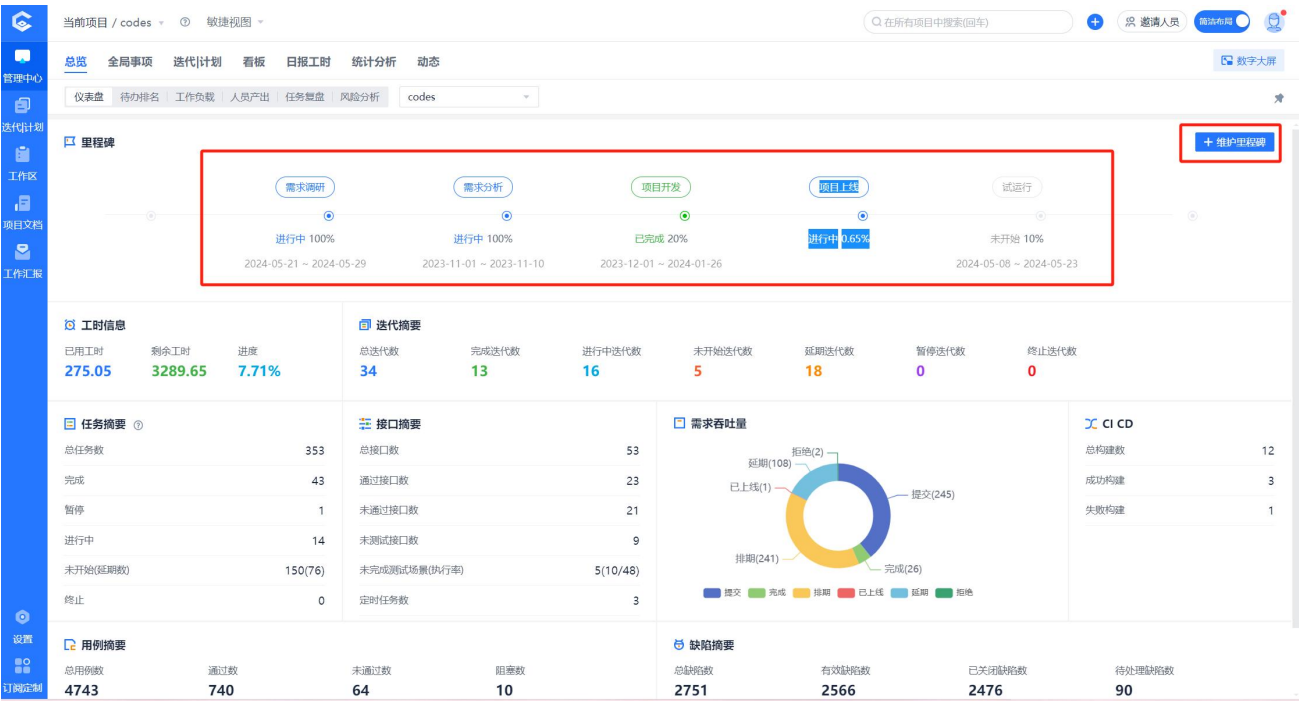


图 2-16

2.21.2.3. 待办排名：

显示某个或所有项目人员待办事项排名，有总排名和各分项排名。第二排各分项排名，人员后的数值点击后会中跳转到全局事项中，且查询人员为待办排名分项中的人员。



图 2-17

2.21.2.4. 工作负载

整合了日历和甘特图，左则为人员。点击人员前的向下箭头图标可展开任务明细。日历上红色数值，说明这一天事项数过多。左上的待排期点击后可批量排期，批量排期时可以倒排也可以顺排。

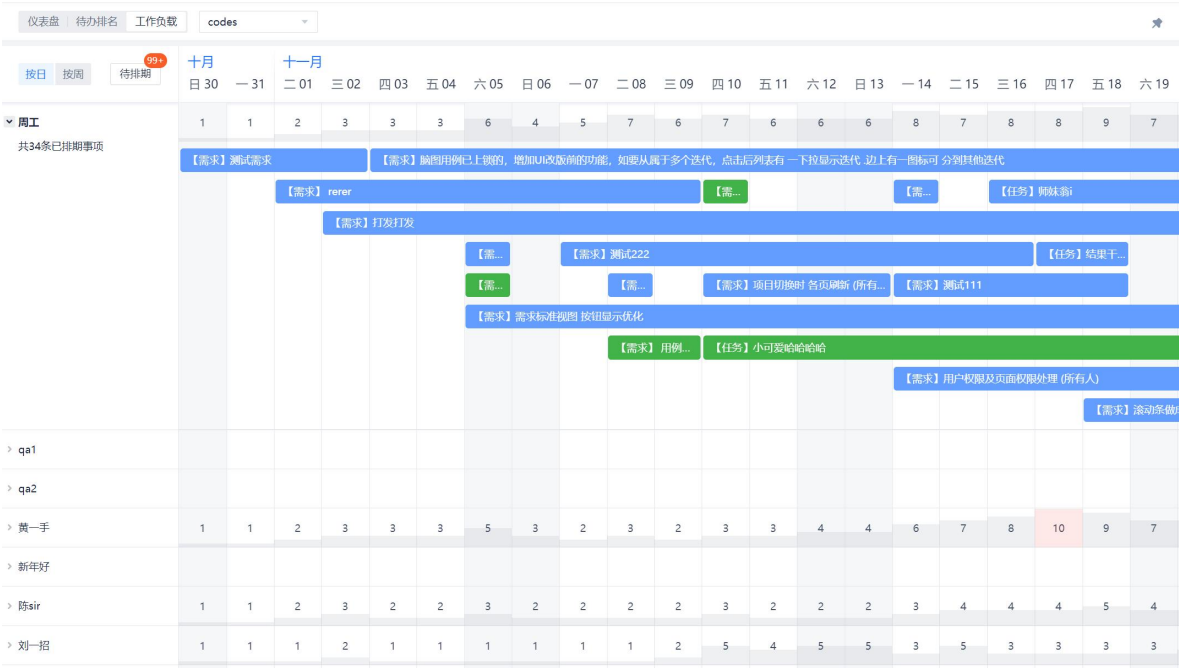


图 2-18

2.21.2.5. 任务复盘



图 2-19

2.21.2.6. 人员产出

按开发，测试和产品一个职位统计时间段内的产出，缺省为最近 7 天，最多可选 3 个月内的时间进行统计。不同职位统计的项目是不一样的，产品统计需求相关，测试统计测试相关工作事项。

提示：只有设置了上述三个职位的人员才能统计到。

产品				
测试				
开发				
日期	姓名	工时(h)	处理缺陷数	处理任务数
2023-12-27	黄一手	0	0	0
	刘一招	0	1	0
2023-12-25	黄一手	0	7	0
	刘一招	0	0	0
2023-12-22	黄一手	0	0	0
	刘一招	0	2	0

图 2-20

2.21.2.7. 全局事项

全局事项下有：待批、需求、任务、缺陷、用例、今日事项。全局事项上的今日事项显示所有人当天

的今日事项。今日事项为有两个来源，1 填写日报时明日计划的事项，2 或是每个人在”我的待办事项“中勾选的今日事项。需求、任务、缺陷、用例都可有不同的视图切换，如看板、甘特图等，不同的事项视图不一样。






图 2-21

2.1.0GAu1 及其后版本中 PM 可以在全局事项下，需求和任务以及看板中，可更改需求和任务状态，其他地方都只有项目参与人可以更改。

2.21.2.8. 管理中心查看迭代计划

以表格的形式显示迭代计划，在右边操作列中可查看代报告和下载迭代报告。且进度双击时会得新计算。

总览 全局事项 迭代计划 看板 日报工时 统计分析 动态							
请输入迭代名称+回车键 Q codes 表格 视图							
迭代名称	所属项目	状态	执行版本	开始时间	结束时间	进度	操作
check status	codes	未开始	v2.0	2024-09-19	2024-09-27	0%	  
new it	codes	未开始	v2.0	2024-09-01	2024-09-13	14.320%	
验证用例待处理事项的BUG	codes	完成	v2.0			18.798%	

2.21.2.9. 工时日报、看板、统计分析

管理中心还有工时日报、看板、单项目的统计分析，跨项目的统计后续增加企业级统计分析功能，面

向公司中层的或企业级的管理人员。

2.21.2.阶段 | 规划或迭代 | 计划：

敏捷视图时显示为**迭代 | 计划**；**瀑布视图**时显示为**阶段 | 规划** 详见 [Codes 敏捷与瀑布的真正融合](#)。

在下图红框标示的地方切换视图。



图 2-22

2.21.2.1. 阶段 | 规化

.阶段可在项目中通过选择模板来生成，也可手动维护，或自动生成后再手动维护，只是只有 PM 在管理中心仪表盘中可以手动维护。阶段侧边栏的右边，点击向右的箭头，可展开其下计划，且点击。。。可创建和维护其下迭代计划。



图 2-23

2.21.2.2. 迭代 | 计划

Codes 采用多事项迭代，从需求到测试到上线 形成闭环。从需求到开发任务，到测试，到交付物到上线。迭代可根据需要按版本创建也可以按时间周期创建。以时间周期创建迭代时，对应的版本是同一个或多个。某些项目可能都是实施的多，可以不同的客户建不同的迭代，每个迭代对应不同客户。详见[多事项闭环迭代](#)。

需求评审后，规化到迭代中，然后开发人员拆分需求为任务，同时测试可写测试用例，过程中的产出放到迭代的交付物中，自动和项目文档关联，上线前在发布中定义 线事项及执行人，可以把发布理解为上线的事项清单，且有先后顺序。在迭代下，把需求分解为子需求，分解为任务，和用例时，他们也会自动分配到当前需求所属迭代下。

当迭代状态设置为完成时，会在自动把当前迭代的报告在项目文档中保存一份。迭代下需求、用例、任务和缺陷还可切换到不同的视图如看板，状态等。



图 2-24

如上图所示，需求分配到迭代下之后，开发人可分解需求为开发任务，点操作栏最后的图标可以创任务，且创建的任务会自动分配到当前迭代下；操作栏第二个图标为创建测试用例，创建的用例也会自动分配到当前迭代下。

提示：如分配需求到迭代前，所分配的需求下就创建有了任务，或用例，会自动把这些任务和用例分配到当前迭代下。

迭代下任务 TAB 中分配任务，这里分配任务是分配没有所属需求的任务。有所属需求的任务，随需求分配到迭代中，不能单独分配有所属需求的任务到迭代中。

迭代下的缺陷 TAB 中,新提交这个 checkbox 指，只查询当天迭代发现的缺陷。

2.21.3.工作区：

面向基层执行人员办理事项的区域，可以处理需求状态，简单分析需求为任务；更改任务状态，创建任务对应的缺陷；处理缺陷，和执行用例。详见[工作区说明](#)。切记处理或流转缺陷时，要点击缺陷状态，而不是点击缺陷描述。

2.21.3.1. 我待办的事项

如工作区相关泳道上的功能不能满足业务需求，可点击各泳道右上的中跳转图标，跳到对应的功能中心进行办理。各泳道下可按输入条件或选择状态进行查询。

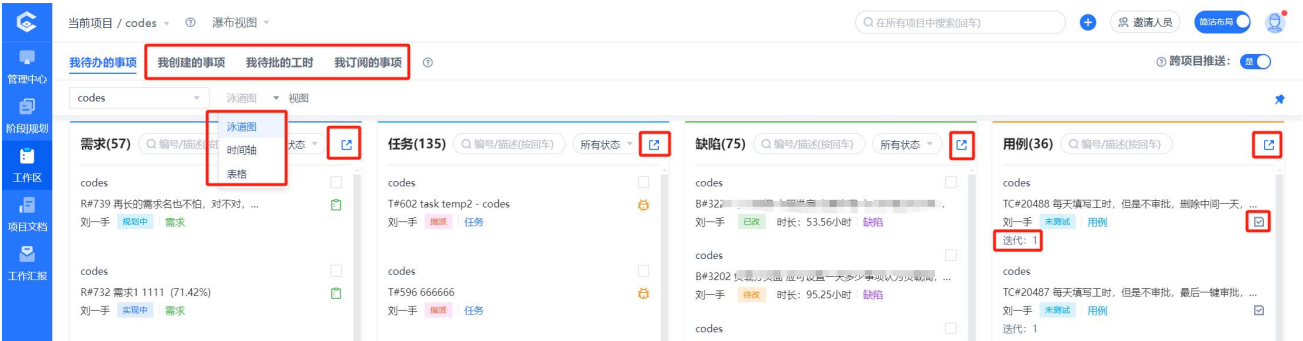


图 2-25

提示：在上图各泳道或是列表中的事项中，每天早上设置今日事项（也可以在日报中填写明日事项）在快下班的时候，填写工时日报。如果任务没开始填写了工时后自动变为开始状态，如剩余工时为 0，就设置任务为完成状态。如手动改需求下的任务，状态，需求不会联动，只有通过工时，需求的状态才会和任务的状态联动。

2.21.2.3. 我创建的事项

展示我创建的事项，切换到不同事项时可以创建对应的事项，迭代可以按版本创建也可以按时间创建。右边跳转图标，跳转到相应功能中心，我待办的事项泳道右上的转跳。



图 2-26

2.21.2.4. 我订阅的事项

可以以时间轴和泳道显示订阅的事项，双击描述可弹窗显示事项详情。

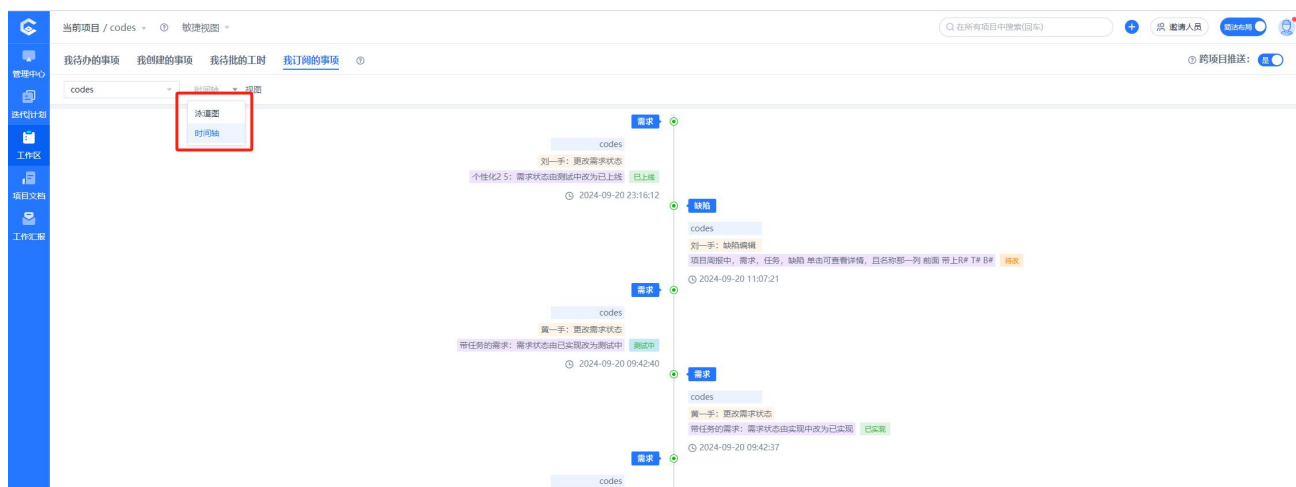


图 2-27

2.21.2.5. 我待批的工时

如果打开了工时日报审批的开关，如果名下有待批的工时日报就是在工作区的这里审批，审批时不存在不通过，审批时主要是可以更改提交的工时信息。

2.21.4.项目文档:

建项目时会为每个项目生成一个项目名为一级目录的项目文档，下面有缺省的目录，可再自行创建。主要功能就是管理上传到 Codes 中的文档，以及外部文档链接，2.2.0 将实现项目原型的管理。项目文档有两个视图，缺省为当前**项目视图**，还可切换到**全局视图**，全局视图时管理人员可以查看所有项目的文档，普通人员只能查看所参与的项目文档和全局视图下创建的公共目录下的文档。迭代交付物也会在项目文档中挂载一个迭代交付物的目录。同时还挂载了一个接口文档的目录，点击会跳到到接口文档中。

在项目视图下创建的目录，为项目的目录；在全局视图下创建的目录为公共目录。且 linux 版本还支持文档预览，引用 kkfile4.1.0 作为文档预览组件。

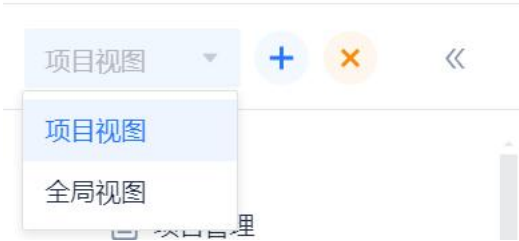


图 2-28

提示：如有已归类整理好的项目文档，可以以 zip 导入的形式批量导入到 Codes 中，会在上传时所点击的节点下，自动建 zip 包内的目录，然的把 zip 包里相关文件放到对应目录中。迭代改为完成状态时，也会在项目文档总结下生成一份迭代报告的文档。

2.21.5.工作汇报：

如图所示：把日报、周报集中放一起组成工作汇报。



图 2-29

2.21.5.1. 日报：

2.21.5.1.1. 个人日报

填写日报时，如是补某一天的日报，那么今日事项，不能补选，查到所补的那天留过痕的数据（前一天日报设置为明日计划的事项、或那天手动设置为今日事项的事项、以及那处改过状态的缺陷、执行用过

例，更改过状态的需求和任务)

Codes 日报和工时聚合一起，比单个填方便很多，每天下班前一分钟便可搞定日报及工时。日报事项中会自动列出当前处理过的缺陷，任务及需求。填写工日报时，最好的作法是也勾选了明日计划，方便领导知道大家明天的工作事项安排，且勾选了明日计划后，第二天填写工时报的时 就算有一些事项当日没有在 Codes 上变更其状态，也会自动列在日报的今日事项中，否则只能以补选的方式来选择当日工作事项。

如果写工时日报时，没有勾选明日事项，也可以在第二天早上，在工作台我的事项下勾选今日事项以让领导明白个人当日的工作计划。

工时日报缺省是要项目的 PM 或部门负责人审批，如不审批，提交日报后不能修改。填写日报时，如有风险还可关联风险事项，临时参与的工作，如开会等可以以其他事项来填写。

提示：需求只有没有子需求也没有任务时 才可以填工时，当需求很简单不用分解为子需求或任务时可以把需求当任务用，也就是只有这样的需求才能填写工时。还可在日报中关联风险。

填写工时，如剩余工时为 0 则认为进度是 100%

2.21.5.1.2. 项目日报

不用填写，自动从个人日报中按项目汇总，且又按人分组来生成。

2.21.5.2. 个人周报（标准版功能）

如果补交只能补上周的周报，上上周的周报是不可以补填的，但如果之前何了草稿就没有上上周的限制。提交后就能再再修改，可以先存为草稿。提交周报时，会检查是否设置了周报接收人。



图 2-30

个人周报分为 4 个部分：**概述、本周工作事项、本周计划及完成情况、下周计划**。这 4 部分中，只有概述中“问题 | 经验 | 及其他”需要输入，和下周计划要勾选之外，其他全是自动生成，如有需要再填写备注。**（注意：必须提交日报，才能周报中自动生成大部分数据）。**

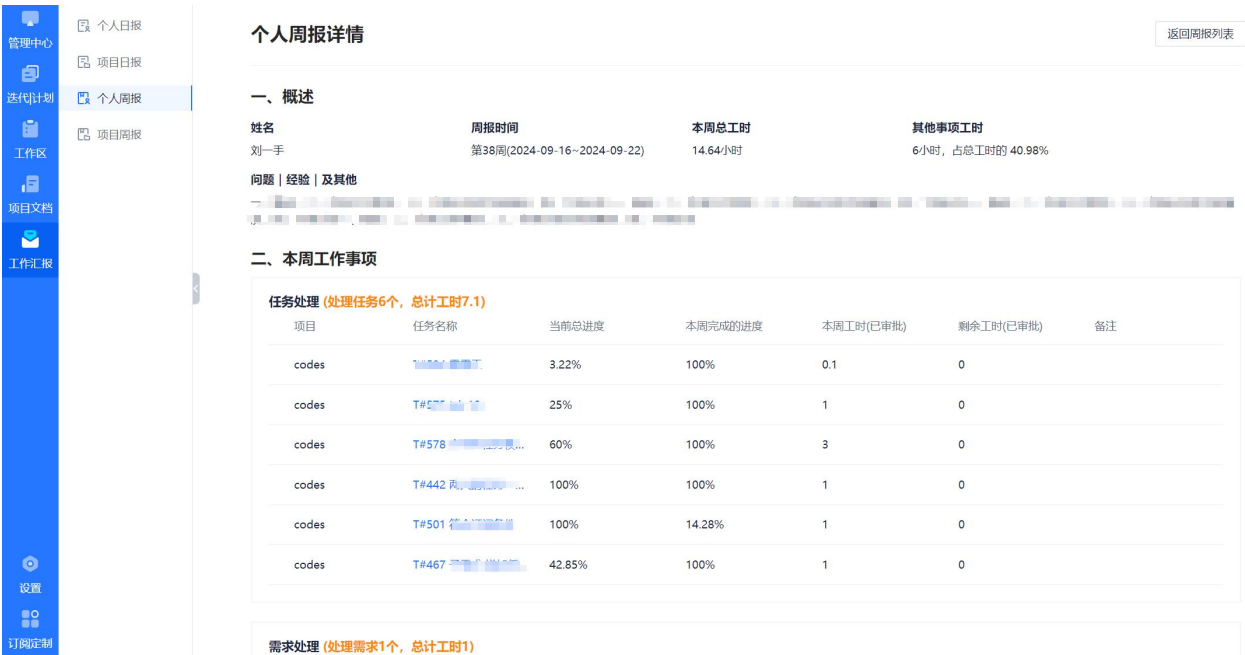


图 2-31

2.21.5.3. 项目周报（标准版功能）

项目周报规则同个人周报，只是数据上有不同。项目周报含 4 个大部分：**项目概述、本周工作事项、项目总体情况、下周计划**。除了下周计划要勾选以及“问题 | 经验 | 及其他”要手写外，其他都是自动

生成。**（注意：必须提交日报，才能周报中自动生成大部分数据）。**

2.21.5.3.1. 项目概述

项目总体进度和“问题 | 经验 | 及其他”，其他都是自动生成

项目周报

保存并提交存草稿取消

一、项目概述

项目名称codes

项目总体进度8.08%

本周工作总工时13.64

本周其他事项工时5 (36.66%)

周报时间第38周 (2024-09-16~2024-09-22)

项目总体进度有所延期

问题 | 经验 | 及其他

图 2-32

2.21.5.3.2. 本周工作事项

包含本周计划完成情况、人员事项完成情况、本周事项详情:、本周质量情况、本周人员质量情况几部分。

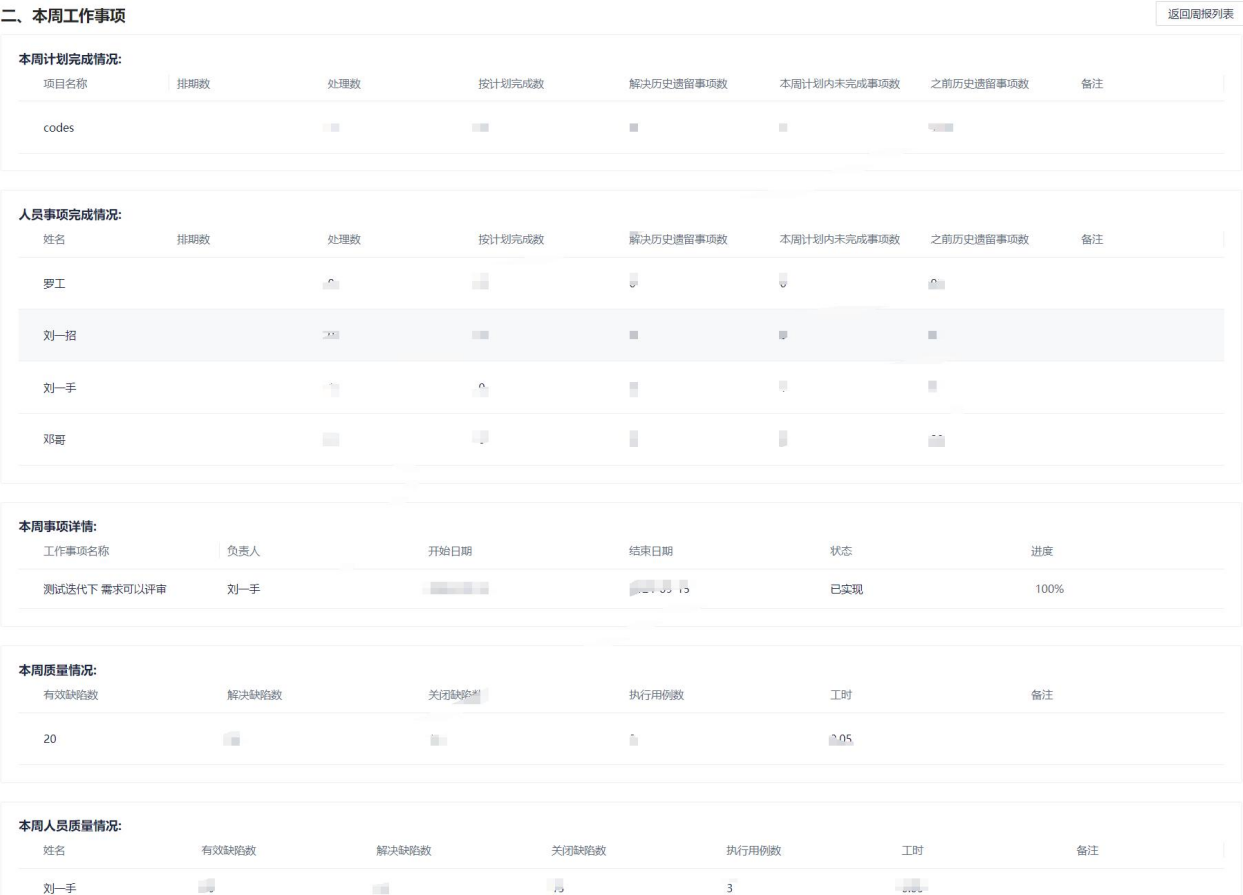


图 2-33

2.21.5.3.3. 项目总体情况

包含里程碑概览、工作事项概览、缺陷修复情况这 4 部分



图 2-34

2.21.5.3.4. 下周计划

包含需求、任务、缺陷的处理计划。

2.22.需求管理

从工作区，我待办的事项下，点击需求泳道的跳转图标中可进入需求管理中心，也可点击工作区我创的建事项下需求 TAB 中跳转图标后进入。



图 2-35

2.22.1.提示

需求如其下有任务，再建子需求时，会把其下任务分到第一个子需求中；如一个叶子需求填了工时，再为其增加子需求时，工时由第一个子需求继承。只有叶子需求可以增加任务。需求的进度由其下任务进度或叶子需求的工时向上推导生成与其自身状态无关。待完成需求是按状态进行查询。

创建需求时如勾选了启用任务模板，会生成两个测试任务如对应的需求分类配置了任务还会自动生成任务模板中的任务。

产品人员（项目中指定为产品人员），进入需求页面时缺省显示产品视图（只显示需求及子需求），其他人显示全景视图，也就是需求下显示任务，用例。

需求分类可以在字典中维护好需求分类，创建项目时会从字典中拷贝一份新的需求分类到项目中，项目中也可自动维护需求分类，并拖动调整分类的关系。

2.22.2.功能说明

需求输入、需求拆分、需求评审、需求排期、需求变更、进度跟踪、。配置需求任务模板后，在需求评审过后可以按模板生成研发任务。。

2.22.2.1. 需求管理中心主界面



图 2-36

2.22.2.2. 创建需求

标准流程时需求只有提交评审后方可进入规划中，精简流程的需求提交后不需评审就是规划中状态。

创建新需求时如果选了标准流程，同时选择了会签评审人员会自动提交评审。



图 2-37

2.22.2.3. 评审及变更

需求状态如果是规划中或者进行中，提交需求更改必须进行变更评审且通过后才能生效，否则其他人看到的是变更前的版本。且变更了会有提示进行提交评审，创建新需求时如选择标准流程，同时选择了会签评审人员会自动提交评审。

2.22.2.4. 需求追踪

点击需求名称，可跟进围绕需求的所有研发活动。

详情-#231 发票管理

基本信息 评审记录 变更记录 缺陷 用例 任务及干系人 流转历史 接口文档 接口用例 关联需求

申请人	申请说明	申请状态	操作
 刘一手	创建需求	通过	
评审人	评审描述	评审状态	评审时间
刘一手	可以排期	通过	2023-03-12 17:23:25

图 2-38

2.22.2.5. 需求分解为任务

在需求规划后，可以分解为开发任务，也支持按需求任务模板按任务模板自动分解任务。Codes 中只要需求的状态变为规划中，不管是否配置任务模板，都会生成两个测试任务，只要在需求管理的页面创建用例，自动会把用例分配到所创建的测试任务中。如需求分配到迭代下，其下的任务也随需求分配到迭代下。

2.22.2.6. 需求工时及进度

Codes 中当需求下有任务时，则这个需求不能填写工时，只能填写该需求下任务的工时，然后通过任

务来计算需求的工时；需求中有子需求时，也只能填写子需求的工时，系统会自动计算父需求的工时。如果当需求很小的时候，建议可以把需求当任务用，而不需要把需求分解为任务。

2.23.用例管理

通过主菜单上的测试进入用例管理，包含：脑图用例维护和用例库双向同步，用例在线执行，离线执行，各种用例视图切换。Codes 以迭代的方式来组织测试用例的执行，执行人员的用例都在同一迭代下，只是不同的人执行不同的用例，以从全局的角度来掌握测试进度。

2.23.1.主界面

从工作区用例泳道中，点击跳转图标进入用例管理中心，或在工作区我创建的事项下用例的 TAB 中跳转图标进入。

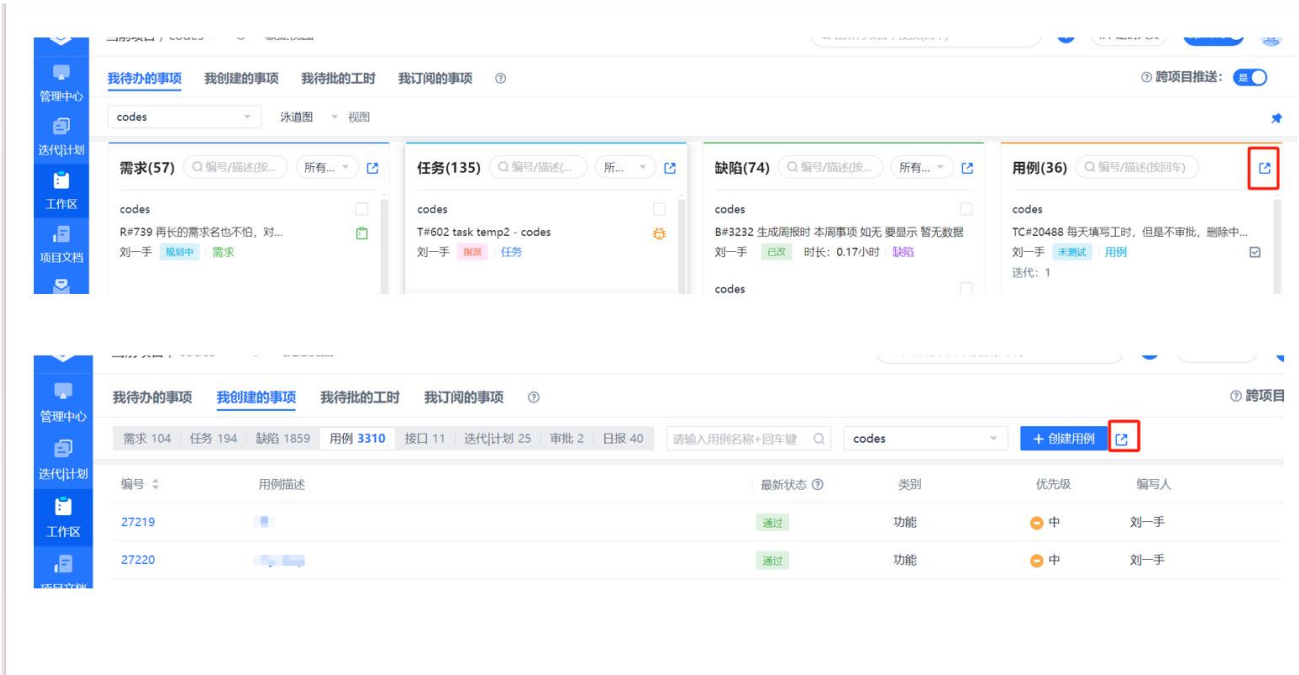


图 2-39



图 2-40

2.23.2.产品用例库和公共用例库

关键在于产品用例库，产品用例库和项目中的用例可以双向同步，且同步时项目需求也一起同步了。

如有变更会自动保存用例历史版本。

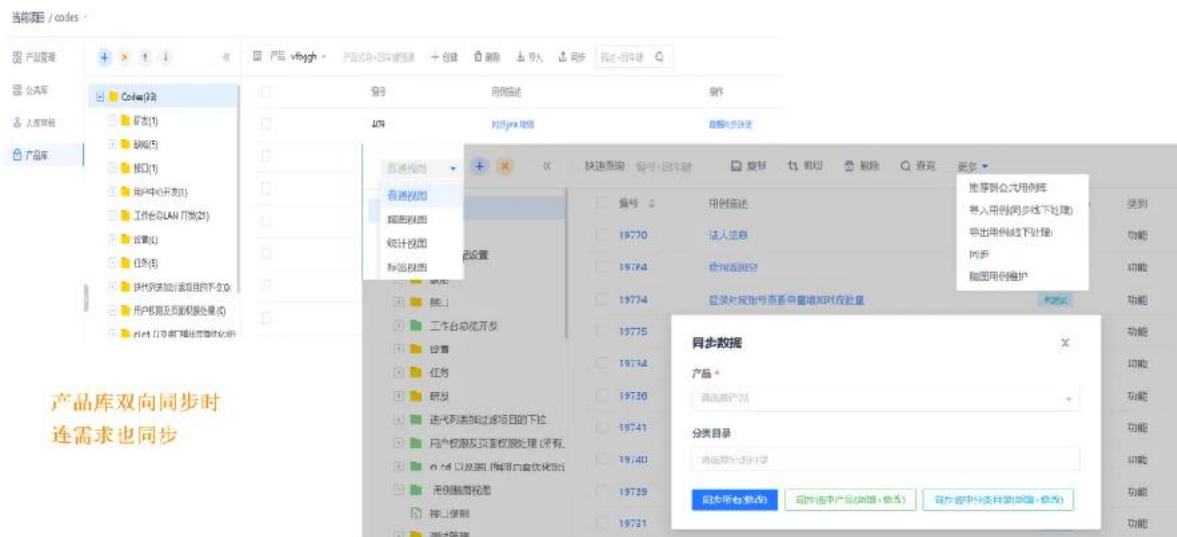


图 2-41

2.23.3.线上线下载同步及导入



图 2-42



图 2-43

导入用例时如输入模块，类型，优先级，标签不存在，会自动在系统上创建。

4743 通过数: 720 未通过数: 67 阻塞数: 11 未测试数: 5934 不通过数: 1; 导出用例数: 201									
codes									
功能项/目录	用例描述(必填)	前置条件	操作过程及数据(必填)	预期结果(必填)	线上状态	离线执行状态	执行版本	项目PM 刘一子	类型(必填) 优先级(必填)
11ms改造/版本属性	物料属性、设备属性、申请单属性、产品属性、检测项目属性、原始记录属性、参考原型图		从xmind 中导入用例数据	从xmind 中导入用例数据	未测试	未执行		功能	中
11ms改造/11ms分类/松源项目详情设备附件表格/设备附件表格/设备附件表格	设备附件表格、设备附件表格、设备附件表格		从xmind 中导入用例数据	从xmind 中导入用例数据	未测试	未执行		功能	中

图 2-44

2.23.4.脑图用例

脑图用例保持脑图的简洁，不用指定特殊的格式，保持脑图的本真用法。叶子节点是用例，其他作为模块。脑图也是以文件来维护，只是转标准用例后，两边双向同步。

2.23.5.业务场景用例

把多个单一用例编排组合为业务场景，也是 Codes 复用用例的另一种方式，且按业务场景来管理，一目了然。除了单一的功能用例，还有不同的业务场景，特别是流程类功能测试，有了场景不怕漏了，还便于管理。

转帐业务场景								
编号	模块名称	用例描述	顺序	执行状态	类别	优先级	编写人	成本
19668		登录	1	未测试	功能	高	刘一手	1
19667		查余额	2	未测试	功能	高	刘一手	1
19666		发起转帐	3	未测试	功能	低	罗工	11
19665		从转帐历史中查...	4	未测试	功能	低	罗工	11
19664		超额转帐	5	未测试	功能	低	罗工	11
19663		正常转帐	6	未测试	功能	高	罗工	11
19657		非实进转帐	7	未测试	功能	高	罗工	2

图 2-45

2.24.执行用例

用例必须分配到迭代下才能被执行，不管是单一的用例还是场景用例都必须在迭代下才能执行。

推荐在迭代中执行用例，如不在迭代中执行用例，需要选迭代才能执行。脑图中也可以执行用例，前提是脑图用例转为标准用例后才能右键执行用例。

2.24.1.执行用例过程

以迭代为中心来组织和开展测试工作

建好迭代后，在迭代下分配用例，如下图所示先分配用例，再分配执行人，最后执行人点击用例状态执行用例。回归测试时可以点快速执行视图，以批量执行用例，也可以在更多下批量通过，批量不通过，批量从迭代中移除已分配到迭代中的用例。也可导出线下执行后，再（导入）同步到线上

codes1.2

需求待修复[回归缺陷]任务测试用例功能业务场景接口场景新提交缺陷场景内用例

分配用例分配执行人执行人所有执行人导入导出用例执行结果

编号	用例描述	状态	类别	优先级	模块名称	预分配人	成本	编写日期
19611	刚刚创建的	未测试	可用性	低	项目	黄一手	6	2022-11-04 09:10:34
19613	功能操作简便	未测试	功能	中	易用性	黄一手	1	2022-11-04 15:43:16
19614	页面布局美观正常, ...	未测试	功能	中	易用性	黄一手	1	2022-11-04 15:43:16
19615	提示语清晰理解	未测试	功能	中	易用性	黄一手	1	2022-11-04 15:43:16
19616	展示相关数据表格及...	未测试	功能	中	页面布局文案及UI样...	黄一手	1	2022-11-04 15:43:16
19617	进入管理页面, 展示...	未测试	功能	中	页面布局文案及UI样...	黄一手	1	2022-11-04 15:43:16
19618	展示该列表数据, 可...	未测试	功能	中	输入存在名称	黄一手	1	2022-11-04 15:43:16
19619	查询返回空	未测试	功能	中	输入不存在得名称	黄一手	1	2022-11-04 15:43:16
19620	查询返回全部数据	未测试	功能	中	输入为空	黄一手	1	2022-11-04 15:43:16

共有 178 条数据12345...209条/页

图 2-46

2.24.2.执行用例说明

执行用例时，左边的“目录树”为当前执行人所分配的用例所对应需求，并显示用例数和执行数，而且可以导出离线执行再导入同步回来。

需求树执行视图

测试需求

codes(178/0)

- 登录(18/0)
- 缺陷流程设置(16/0)
- 缺陷(33/0)
- 接口(30/0)
- 工作台总览开发(21/0)
- 任务(12/0)
- 研发(1/0)

编号	用例描述	状态	类别	优先级	模块名称	预执行人	成本	编写日期
19610	李	未测试	可用性	高	项目	黄一手	5	2022-11-03 ...
19611	刚刚创建的	未测试	可用性	低	项目	黄一手	6	2022-11-04 ...
19613	功能操作简便	未测试	功能	中低	易用性	黄一手	1	2022-11-04 ...
19614	页面布局美观...	未测试	功能	中	易用性	黄一手	1	2022-11-04 ...
19615	提示语清晰理...	未测试	功能	中	易用性	黄一手	1	2022-11-04 ...

总用例数: 178 通过: 0 未通过: 0 阻塞: 0 不适用: 0 未测试: 178 进度: 0%

图 2-47

用例执行结果

总进度

总用例数192

通过数0

未通过数0

阻塞数0

不适用数0

未测试数192

冒烟通过率0%

进度0%

执行进度

分配人	总用例数	通过数	未通过数	阻塞数	不适用数	未测试数	进度
黄一手	145	0	0	0	0	0	0%
刘一手	33	0	0	0	0	0	0%

图 2-48

2.24.3.脑图用例执行

如果只用脑图用例，直接把脑图文件分到一个或多个迭代下，然后直接在脑图中执行用例，只是在执行前先切换到在哪个迭代下执行即可，从用例管理中心脑图维护中进入。可以导入 Xmind，也可直接在 web 编写脑图。



图 2-49

2.24.4.用例快速执行

如果是回归测试或是对测试用例很熟，可以批量快速执行，如下图所示，左边的树上显示当前执行人所分配用例对应的需求，及各需求下用例数和执行数，如执行完会有一个勾。当然也可导出后线下执行再同步到线上。



图 2-50

2.25. 缺陷跟踪

在任何缺陷列表(迭代下, 工作台下, 缺陷管理列表下) 点击缺陷状态, 流转缺陷。

2.25.1.缺陷管理中心主界面



图 2-51

2.25.2.缺陷流转

点击缺陷状态流转缺陷，切记不是点击缺陷描述。

在 Codes 中处理缺陷，无需关心缺陷有多少种状态，缺陷控制引擎会自动根据测试流程，缺陷当前状态及处理人员在项目测试流程中所处的流程节点自动计算，当前可转换为什么状态，并自动流转关联节点作为下一处理人。在缺陷管理列表中，点击某个缺陷的状态进入缺陷处理。



图 2-52

正常情况下重分配用于经分析这缺陷不是我的，通过重分配转其他人。也还有一个非正常用法：如果一个缺陷需要多人来修改，可以 Dev A 修改后以重分配的形式转 Dev B，Dev B 修改后再转 Dev C。

2.25.3.gitLab 提交代码时和 Codes 缺陷关联

前提配置好 gitLab 上的 webhook，配置参见 [gitLab 提交代码和 codes 里 bug 关联配置](#)

在 gitlab 上提交代码时，注释中 B 加缺陷编号，如 B3855，可以一个#号后加多个缺陷 id，多个缺

陷 id 间用空格隔开如 B124 456, 也可多个, B123 B345, 关联后, 缺陷的 cmmmit 历史如下图所示。



图 2-53

缺陷和 commit 关联后, 描述显示为棕色

我待处理的

我提交的

我修复的

我关闭的

我参与的

全部

快速查询: bug编号+回车

bug描述+回车键

刷新

+ 增加

复制

删除

导出

查询...

编号	bug描述	状态	等级	类型	优先级	测试人员	开发人员
<input type="checkbox"/> 3990	CommitIDTest3	待改	P1	功能	中		
<input checked="" type="checkbox"/> 3989	CommitIDTest2	待改	P1	功能	中		
<input type="checkbox"/> 3988	CommitIDTest1	待改	P1	功能	中		

图 2-54

2.26.接口测试流程及接口 Mock

基本流程: (1)BaseUrl 设置-----> (2)基础认证设置 -----> (3) 维护接口用例----->(4)建接口测试场景(可在迭代中直接增加)--->(5)手动执行接口测试场景(可单个,也可一键执行场景中所有接口)或定时执行测试场景。

Mock 接口, 访问 URL 和 Codes 的 IP 一样, 只是 http 端口为 8064 +Mock 中的 url;

2.27.压测流程

基本流程：（1）上传脚本----->（2）维护 jmeter 节点 ----->（3）创建压测场景----->（4）执行测试场景--->（5）查看测试报告。

2.28.工具触发执行接口测试

Codes 有对外接口, 比如在 jenkins 中 build 完成并部署后, 在 job 中或量 beanshell 中通过 curl 执行, Codes 上的定时任务的测试场景, 在配置 curl 时, 需要带上在 Codes 上配置的安全访问 key 。除了配置访问的 key 也就是 token 外, Url 中还要带上要实时执行的 Codes 上的定时任务的编号, 可以是多个用带号分开, 或是传 taskId 则执行所有测试场景。Url 如下所示 : http://codes 服务器 IP:端口

/codesBackend/jenkinsCallBackCodes?token=XXXX&ids=xxx,xxx,xx 或 http://codes 服务器 IP:端口

口/codes/jenkinsCallBackcodes?token=XXXX&taskId=xxx

token 为系统管理下, 安全访问 key 中配置的启用状态的 jenkins key ;传 taskId 时跑所传项目 id 下的该项目的接口自动化定时任务, 传 ids 时, 跑指定的 1 个或多个定时任务, 多个定时任务编号用逗号隔开。

2.29.CI CD 流水线编排

只有标准版（如使用基础版, 需要准备 Docker 容量并联系我们提供组件进行安装）才有 CI CD 功能, 需要先在 Codes 中创建好代码库, 然后构建流水线, 以拖拉的形式编排流水线。

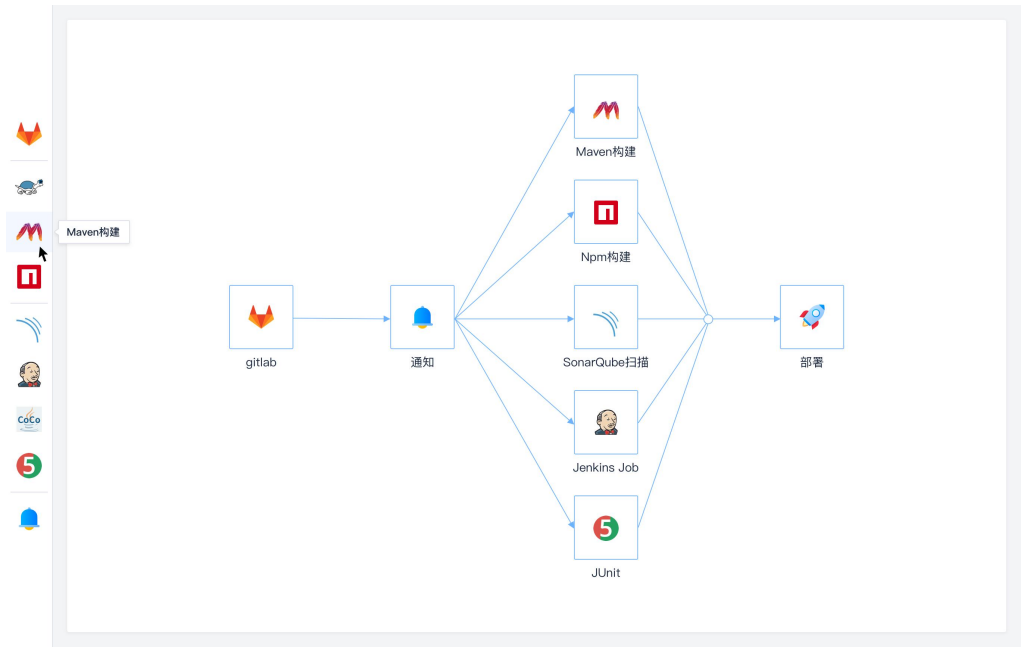


图 2-55

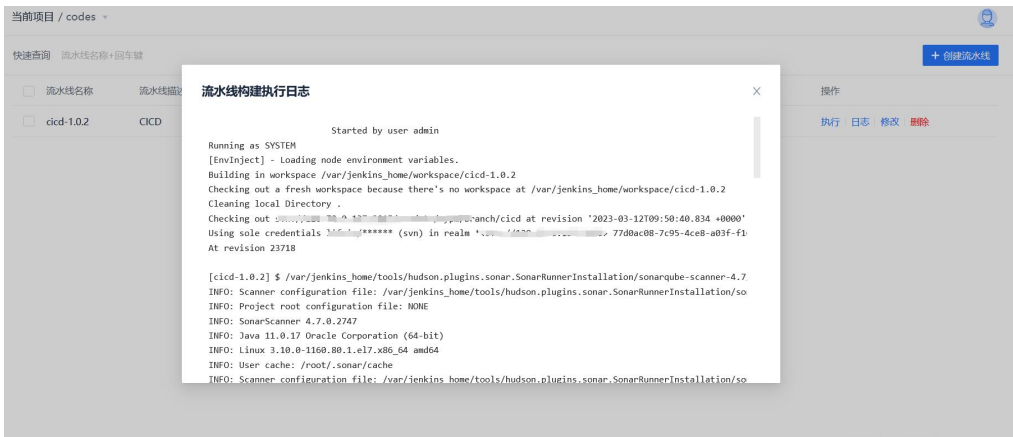


图 2-56

2.30.任务管理

Codes 中有两种类型的任务，一种是和需求关联的标准任务，另一种是独立的不关联需求的任务，独立任务缺省是要走审批流程，详见[任务流程设置](#)。和需求关联的任务，不能直接分配她到迭代中，只能分配他所属的需求到迭代中，然后需求下的任务会随需求分配到上应的迭代中。在迭代下任务 TAB 中分配任务时，就是分配的独立任务，在迭代下任务 TAB 中创建的任务也是独立任务。在迭代下需求 TAB 分解需求

为任务时，会自动把分解的任务也分配到迭代中。

2.31.原型管理

中有两种类型的任务

3、主要功能使用说明

3.1 需求管理

3.1.1 需求视图及主界面功能

需求管理有多种视图模式，不需要在建项目时选择项目模板，有看板模式，也有甘特图模式还有全景模式以及产品模式，另外还有标准视图，迭代视图，脑图视图，文档视图，方便从不同的维度来维护和查看需求。**如在项目中作为产品人员，则缺省显示产品视图，否则显示全景视图；产品视图和全景视图的主要区别在于，全景视图显示需求下的任务和测试场景，产品视图如果要显示需求下任务，需要在右侧更多中点击展示任务。**需求管理的提示信息见 [2.6.6 需求管理](#)；需求主页如下图所示：



图 3-1

需求可以有子需求，分页是按主需求来进行分页

3.1.2 需求状态及创建需求

需求状态包括：创建中、草稿、规划中、实现中、已实现、暂停、已拒绝、待提交评审、变更评审中；

规划中、实现中、已实现、暂停、已拒绝这 5 个状态时可以点击更改状态。

精简流程不需要进行评审点击保存后就是规划中状态，标准流程创建的需求为创建中状态，只能提交评审后才能变为规划中；如选标准流程且设置了会签评审人员,保存后会自动提交评审。

保存为草稿可以任意修改且不用提交变更，规划中或其他状态的需求如果进行了修改，只能变更评审通过后才能生效。缺省需求层次不能大于 2 级，同时可以在项目维护中进行修改。

创建需求

需求名称 *

请输入需求名称

需求描述

H1 H2 H3 B U I A • • • • •

请输入内容...

精简流程不需要评审保存后就是规划中状态，标准流程创建后的需求是创建中状态要提交评审才能可能变为规划中；如选标准流程且设置了会签评审人员,保存后会自动提交评审

附件 上传文件

保存为草稿可以任意修改且不走变更，规划中或其他状态的需求只要修改了，只能变更评审通过后才生效

请选择需求处理流程

标准流程

精简流程

需求分类 *

通用需求分类/安全需求

迭代

请选择迭代

需求处理流程 *

请选择需求处理流程

参与人

请选择参与人

负责人

请选择负责人

预计开始

请选择预计开始

预计结束

请选择预计结束

标准工作时长(小时)

请输入 -单位1小时

优先级

请选择优先级

确认 保存草稿 取消

图 3-2

3.1.3 需求列表操作说明

需求列表上所有带有问号图标的列，都可以点击。

3.1.4 查看需求拉通的研发活动

在需求列表上，点击需求名称，显示和当前需求的所有内容包括：需求详情、评审记录、变更记录、缺陷、用例、任务及干系人、流转历史、接口文档、接口用例和关联的需求。



图 3-3

3.1.5 查看需求变更历史

在需求列表上，点击需求名称，通过弹窗页面切换到变更记录中，点击对比能查看变更前后区别。



图 3-4

3.1.6 设置负责人

在需求列表上，点击某个需求的负责人列，弹窗页面如下图所示：左侧人员为参与人，右侧从参与人中选择一个负责人，负责人只能唯一。



图 3-5

3.1.7 单个排期

在需求列表上，点击某个需求开始或者结束时间列，可以对单个需求的进行排期或是更改排期。如要批量排期在工作中总览工作负载中点待排期数进行[批量排期](#)。缺省一个需求的粒度不能超过 5 天所以排期的时间间隔不能大于 5 天，可以在项目维护中更改需求粒度和需求层次配置；缺省需求层次不能大于 2 级。



图 3-6

3.1.8 创建子需求及关联需求

在需求列表右边操作列更多中点击关联需求可以新增关联需求；在操作列点击新增以创建当前需求的子需求，缺省需求层次只可以创建二级，可以在项目维护中进行更改。



图 3-7

3.1.9 填写需求进度

需求列表中，规划中、实现中、已实现、暂停、已拒绝这 5 个状态的需求点击修改状态和填写需求进度和更改需求状态。1.2.X 版本后 进度只能通过工时来推算，不能再直接填写需求进度。

3.1.10 创建用例，缺陷及任务

在需求列表右边操作列更多中点击增加用例或增加缺陷，或者增加任务时就可以创建该需求对应的实体对象。

草稿状态的需求只能修改和删除，不能增加其他功能操作。

如果需求下生成了测试场景，在需求列表中为其增加用例，增加的用例会分配到这个需求的场景下。



图 3-8

3.1.11 执行需求下用例

在需求列表中，如下图所示为测试场景的实体，在右边操作列中二次分配用例和执行用例。



图 3-9

3.1.12 填写需求下任务进度

在需求列表中，图标为任务的实体，如果是任务负责人，点击任务状态可填写进度。

3.1.13 评论需求

在需求列表右边更多操作列中点评论可能查看之前的评论和发表新评论。



图 3-10

3.1.14 维护需求分类

在左侧需求分类树上点击图标进行需求分类的维护，也可以在字典配置需求分类字典，创建项目时就会把字典中的分类 copy 一分到项目中。

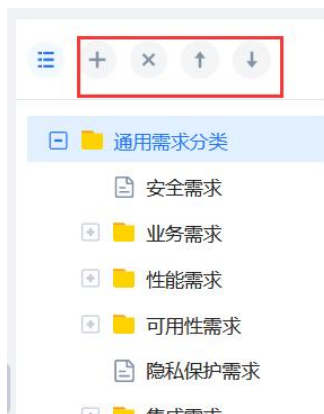


图 3-11

3.1.15 需求分类树切换统计模式

在左侧需求分类树上点击上方最左侧图标可以切换到统计模式，该模式会显示分类下需求数。

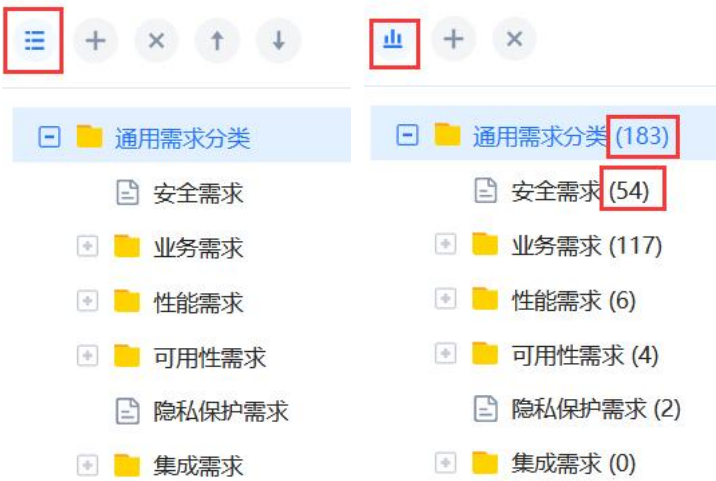


图 3-12

3.1.16 自动生成需求任务

在任务模板中配置好模板后，在需求变为规划中状态时，会按模板对应的需求分类的配置生成任务。
一个分类可以有多个模板，同一时间内只能有一个是启用的默认模板。

新增

任务名称 *

请填写任务名称

任务顺序

请填写任务顺序

负责人

请选择

开工前产出物

请选择

交付产出物

请选择

准入指标

请选择

准出指标

请选择

确定

取消

图 3-13

3.1.17 gitlab 代码提交与需求关联

前提配置好 gitLab 上的 webhook，[配置参见 gitLab 提交代码和 codes 联分理处配置](#)

在 gitlab 上提交代码时，注释中 R 加需求编号，可以一个 R 后加多个需求 id，多个需求 id 间用空格隔开如 R124 456，也可多个 R，R123R345，关联后，需求的 cmmmit 历史如下图所示。

	提交人	备注	提交日期	url
1			2022-03-11 14:58:33	https://et/umb
2			2022-03-11 14:56:36	https://1.net/umb

图 3-14

3.2 任务管理

3.2.1 概述

如不进行需求管理，或是想管理和需求无关的事项，可以直接以任务的形式来安排工作，任务可从属于某个需求，也可以不从属于任何需求。如果是需求任务模板自动生成的任务，显示时会列出需求名。可以通过任务模板批量创建任务。

<div>我负责的 我参与的 我创建的 我关注的 所有任务 任务甘特图 + 创建任务</div>										
<div>快速查询 名称+回车键 项目 codes 状态 请选择 请选择日期 已 导出任务</div>										
任务名称	所属项目	任务参与者	任务负责人	用时(H)	剩余(H)	进度(%)	状态	开始时间	截止	操作
制作安装包	codes	刘一手	刘一手				未开始	2023-02-02	2023-02-03	填写进度 详情 修改 删除
定时备份	codes	刘一手	刘一手	10	9	10	进行中			填写进度 详情 修改 删除
数据同步	codes	陈一招,刘...	刘一手				未开始			填写进度 详情 修改 删除
用户管理	codes	刘一手	刘一手				未开始			填写进度 详情 修改 删除
前端开发	codes	刘一手	刘一手			0	未开始			填写进度 详情 修改 删除
用户中心...	codes	刘一手,陈...	刘一手			0	未开始			填写进度 详情 修改 删除
用户中心...	codes	刘一手,陈...	刘一手			0	未开始			填写进度 详情 修改 删除
req task r...	codes	刘一手	刘一手			0	未开始			填写进度 详情 修改 删除

图 3-15

创建任务

任务名称 *

填写任务名称

任务描述 *

请填写任务具体内容

所属项目

请选择

需求

请选择

品

紧急程度

请选择

难易程度

请选择

任务类别

请选择

工作量(H)

请填写正整数

开始日期

请选择

结束日期

请选择

执行人 *

请选择

负责人 *

请选择

关注人

请选择

附件 [上传文件](#)

确定

取消

需求多时,点击图标可以更友好方式显示需求

如人员多时,点击人员头像查询人员

图 3-16

3.2.2 批量排期

在工作台总览中工作负载下，点击待排期数进行[批量排期](#)。

总览 我的事项 全局事项 我创建的 迭代|计划 统计分析 动态

仪表盘 待办排名 工作负载

codes

按日 按周 待排期 99+

十月 十一月

日 30 一 31 二 01 三 02 四 03 五 04 六

> 周工	1	1	2	3	3	3	
< 2021							

图 3-17

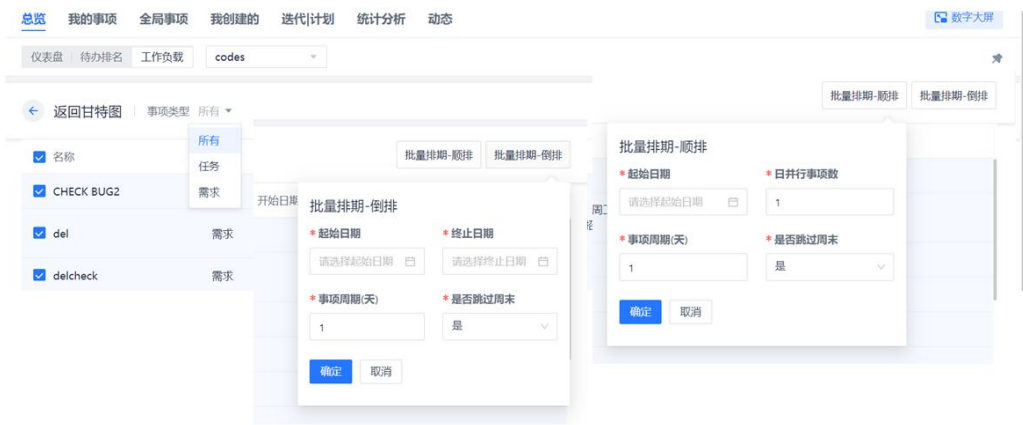


图 3-18

3.2.3 Gitlab 代码提交与任务联联

前提配置好 gitLab 上的 webhook, [配置参见 gitLab 提交代码和 codes 联分理处配置](#)

在 gitlab 上提交代码时, 注释中 T 加任务编号, 可以一个 T 后加多个任务 id 多个任务 id 间用空格隔开如 T124 456 , 也可多个 T, T123T345, 关联后, 任务的 cmmmit 历史如下所示

	提交人	备注	提交日期	url
1			2022-03-11 14:58:33	https://gitlab.com/umb
2			2022-03-11 14:56:36	https://gitlab.com/umb

图 3-19

3.3 测试流程设置及版本维护

3.3.1. 概述

在 Codes 中, 对于每个测试项目, 可以从 Codes 提供的流程中选取部分或全部流程作为当前项目的测试流程, 测试过程中可以实时调整测试流程, 取消某个可选流程时, 系统会把停留在该节点上未处理的缺陷自动流转到下一流程, 并从下一流程节点处理人员中选取一人作为待处理人。

在流程设置中主要完成的工作: 1、设置流程 2、指定在各个流程节点上的相关人员 3、设置被测试软

件版本；另外需要注意一旦在测试流程设置中，指定了测试负责人（可以有多个）。如下图所示，执行顺序为从上到下。在流程设置页面中，切换到“版本维护”，设置软件版本；设置版本时，序号应随版本号增长，不可重复，也不可以和被逻辑删除的版本序号重复；已停用的版本，不在选择版本的下拉列表中列出。修改操作时，可以直拉点三个步骤的表头，直接切换，如点击“项目信息维护”。

步骤	负责人
<input checked="" type="checkbox"/> 提交问题	测试负责人 * (黄一手, 刘一手)
<input type="checkbox"/> 测试交叉	测试人员 * (刘一手)
<input type="checkbox"/> 分析问题	互验人员
<input type="checkbox"/> 分配问题	分析人
<input type="checkbox"/> 分配问题	分配人
<input checked="" type="checkbox"/> 修改问题	修改人 * (刘一手)
<input type="checkbox"/> 开发互检	互检人
<input checked="" type="checkbox"/> 分歧仲裁	仲裁人 * (刘一手)
<input type="checkbox"/> 项目关注	关注人
<input checked="" type="checkbox"/> 测试确认	
<input type="checkbox"/> 通知关注人	
<input type="checkbox"/> 通知指派人	
<input type="checkbox"/> 通知仲裁人	

带星的为必选流程，资深用户可启用可选流程

流节点被check 才认为启用该流程，可实时修改流程

图 3-20

3.3.2. 不同流程使用场景

3.3.2.1. * 提交问题：

必选流程，人员主要为测试人员，不在这一流程节点上的人员也可填报缺陷，但不在这一流程节上的人员不能关闭缺陷。

3.3.2.2. 测试互验：

可选流程，当测试人员和研发工人员不在同一地点办公时，或想测试其他人提交的缺陷时，开启该流程，由资深测试工程师来做测试互验，既可以指导其他人编写高质量的缺陷，也可以在研发人员处理缺陷

前，测试人员内部先进行检查新提交的缺陷，省去了可能的因其他人描述理解差异上，或是其他人可复现上带来的和研发人员的沟通成本。

3.3.2.3. 分析问题:

可选流程，分析缺陷产生的原因，估算修复缺陷需要的时间及期限，一般为研发经理、系统分析师来做分析工作。

3.3.2.4. 分配问题:

可选流程，单元测试时，或团队规模较小且测试工程师清晰的知道开发工程师所负责的模块时，可以不启用该流程，测试人员提交的缺陷，直接分配给开发人员。一般分配人应该为研发经理、研发组长等，可以有多个分配人。

3.3.2.5. * 修改问题:

必选流程，此为修复缺陷的环节，设置的人员是研发人员。

3.3.2.6. 开发互检:

此为研发工程师修改完缺陷后，互相之间的交叉检查。设置的人员是研发人员。

3.3.2.7. * 分歧仲裁:

必选流程,当测试工程师和研发工程师对某个缺陷的无法达成共识时,或研发工程师要求缺陷延期修改,或不计划修复某个缺陷时,由仲裁人进行裁决。一般仲裁人为研发经理、产品经理。

3.3.2.8. 项目关注:

可选流程，设置项目关注人员，这些人员，在项目中不做具体的工作，设置为关注人后，这些人员可以在“切换测试项目”中切换到所关注的项目，当然切换后，能查看数据菜单，由具体人员的权限来决定。管理人员，可以查看所有项目，其他人员，只能在“切换测试项目中”，查看到所参与的测试项目。只要设开启了某个流程节点，且以被设置到了某个流程节点，才算参与了某个项目，如果流程节点没有启用，只

是设置了这节点上的人员，也不算参与了该项目。

3.1.17.1 * 测试确认：

必选流程，且不需要再指定确认人，复用提交问题环节的测试人员，确认为验证缺陷是否可以关闭。

3.4. 缺陷管理

3.4.1. 概述

Codes 里，进入缺陷管理时，缺省显示[我待处理的](#)，也就是流转到操作者本人节点下的缺陷，要查询其它缺陷，可通过查询的方式去查询。如果测试人员 A 关闭测试人员 B 提交的缺陷，前提是开发人员设置该缺陷为“已改”状态；同理开发人员 A 也可以把开发人员 B 节点下“待改”，设置为“已改”状态；仲裁人，以及分配人，可以处理其它同类人员节点下的缺陷。

点击缺陷状态，弹出缺陷外理页面，可更改缺陷状态，不同状态会流转到不同的节点上的不同处理人。如当前缺陷没流转到本人节点下，点击缺陷状态，显示的是缺陷详情。如果在流程设置中开启了缺陷邮件通知，缺陷流转过程中，下一个处理人，会收到缺陷通知邮件，在邮件中会有缺陷处理的链接，点击就自动登录 Codes，并弹出邮件中的缺陷的处理界面。缺陷如何流转是根据测试流程中设置的流程来决定，不同的流程会对应不同的缺陷状态，缺陷状态演化时，会按测试流程设置中设定的各流程节点上的人员之间来流转，[缺陷跟踪主界面见这里所示](#)。

3.4.2. 新增缺陷

项目内所有人员，不仅仅是测试人员，都可以新建缺陷报告，但只有测试人员可以确认缺陷的修复情况。在新建缺陷页面中，缺陷状态由测试流程自动生成，如下一流程为测试互验，则状态为“待置”且下一处理人为测试互验人；如下一流程为分析则状态为“分析”且下一处理人为分析人；如下一流程分配则状态为“分配”且下一处理人为分配人；如下一流程为修改问题，则状态为“待改”且下一处理人为开发

人员。只有测试人员，和缺陷报告的创建人可以修改缺陷报告。

新建缺陷时，除缺陷描述和再现步骤需手动输入，其他下拉列表形式的必填项，通过在基础字典中设置的默认项，来设置缺省值，如不用缺省值自行修改即可。



图 3-21

3.4.3. 流转缺陷

相关人员处理缺陷时，不用关心缺陷有多少种状态，缺陷控制引擎会自动根据测试流程，缺陷当前状态及处理人员在项目测试流程中所处的流程节点自动算出来，当前可转换为什么状态，以及谁作为下一处理人。在缺陷管理列表中，点击某个缺陷的状态进入缺陷处理。

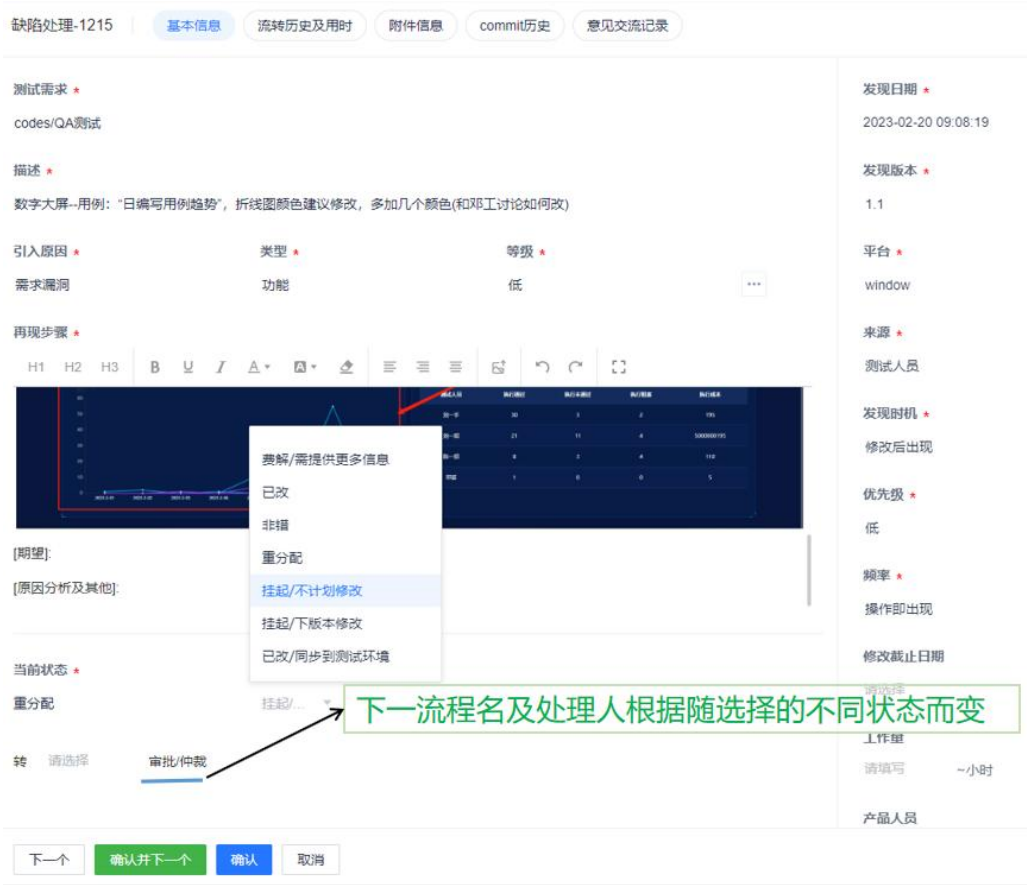


图 3-22

如当前处理为流程为测试互验、分配或分析问题，要求填写校验版本，既验证该缺陷所现在的版本。

在更改状态后，Codes 缺陷控制引擎根据是否需要自动要求，选择下一处理人。如下一流程为分配，更改状态为分配，则在缺陷状态下方会显示，转分配人的输入域，移入焦点到转分配人输入域后弹出分配人供选择；如下一流程人测试互验，则需要选择测试互验人；如把状态更改为“分歧”，则要求选择仲裁人；不是所有状态都要选择下一处理人，如开人员人修改缺陷为已修改，如下一流程为测试确认，则不需要指定下一处理人。

流转缺陷时，可以在再现步骤中加入新内容，提交所做处理后，会自动保存加入的新信息，同时连同处理结果及所做变更记录到处理历史中。

如当前缺陷没流转到本人，或当前缺陷当前状态不是本人所设置，这时点击缺陷状态进入的是缺陷明

细页面。如把缺陷状态修改为“重复”要求关联一个与之重复的缺陷，关联重复的缺陷时，要求被关联的缺陷，报告时间在当前要处理的缺陷报告时间前。

3.4.4. 流转缺陷入口

有 4 个入口：

- 1.管理中心：我的事项、我创建的、全局事项；
- 2.缺陷管理；
- 3.迭代下的待处理缺陷。
- 4.工作区

需求

待修复回归缺陷

任务测试用例功能业务场景接口场景

新提交缺陷

场景内用例

与分配缺陷

已移除缺陷

处理人: 所有+状态: 所有+缺陷汇总

编号	缺陷描述	状态①	等级	流转于	时长(h)	发现人	待处理人	报告日期
<input type="checkbox"/> 132	150%缩放时显示错位	关闭/已解决	中	2022-11-14 16:58...	-	刘一振		2022-11-03
<input type="checkbox"/> 131	预分配人员时 不应...	关闭/已解决	中	2022-11-18 18:45...	-	刘一振		2022-11-03
<input type="checkbox"/> 130	用例库 公用用例库 ...	关闭/已解决	中	2022-11-22 14:38...	-	刘一振		2022-11-03
<input type="checkbox"/> 129	研发下的报表,迭代...	关闭/已解决	中	2022-11-11 15:30...	-	刘一振		2022-11-03
<input type="checkbox"/> 128	修改用例和创建选...	关闭/已解决	中	2022-11-16 16:59...	-	刘一振		2022-11-03
<input type="checkbox"/> 127	增加用例后,没生成...	关闭/已解决	中	2022-11-09 14:48...	-	刘一振		2022-11-03
<input type="checkbox"/> 126	BUG再现步骤富文...	关闭/已解决	中	2022-11-14 17:25...	-	刘一振		2022-11-03
<input type="checkbox"/> 125	项目上的三个快速...	关闭/已解决	中	2022-11-09 14:49...	-	刘一振		2022-11-03
<input type="checkbox"/> 123	提交—上Bug 如要...	关闭/已解决	中	2022-11-14 17:09...	-	刘一振		2022-11-03
<input type="checkbox"/> 121	测试单下的场景 ...	关闭/已解决	中	2022-11-14 17:22...	-	刘一振		2022-11-03
<input type="checkbox"/> 118	项目基本信息页面...	关闭/已解决	中	2022-11-09 14:48...	-	刘一振		2022-11-03
<input type="checkbox"/> 133	SQL服务加载太慢	关闭/已解决	中	2022-11-18 18:45...	-	刘一振		2022-11-04
<input type="checkbox"/> 138	缺陷的富文本编辑...	关闭/已解决	中	2022-11-14 17:10...	-	刘一振		2022-11-04

共有 19 条数据12>14 条/页

点击关联用例

点状态进入流转BUG页面，如不是当前处理人，点击后是查看详情

图 3-23

3.4.5. 缺陷状态集合

因缺陷状态随测试流程而来，且可以实时修改测试流程，所以查询时缺陷状态下拉列表状态会有些多，可以通过在基础字典中维护缺陷状态集合，一个状态集合下包含缺陷状态，缺陷查询的时候，选一个状态集合，代表查询的是这集合内的所有状态。



图 3-24

3.4.6. gitLab 提交代码时和 codes BUG 关联

前提配置好 gitLab 上的 webhook, [配置参见 gitLab 提交代码和 codes 联分理处配置](#)

在 gitlab 上提交代码时, 注释中#号加 Bug 编号, 如#3855 , 可以一个#号后加多个 BUG ID 多个 BUG id 间用空格隔开如 #124 456 , 也可多个#号, #123#345, 关联后, Bug 的 cmmmit 历史如下所示,



图 3-25

3.5. 测试用例管理

3.5.1. 概述

通过主菜单上的测试进入用例管理, 包含: 脑图用例维护和用例库双向同步, 用例在线执行, 离线执

行，各种用例视图切换。Codes 以迭代的方式来组织测试用例的执行，执行人员的用例都在同一迭代下，只是不同的人执行不同的用例，以从全局的角度来掌握测试进度。



图 3-26

3.5.2. 功能主界面



图 3-27

3.5.3. 总览

显示当前测试执行情况。

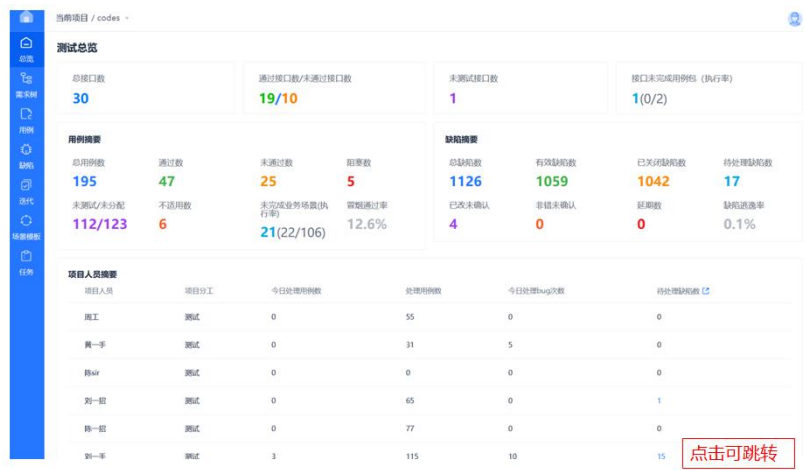


图 3-28

3.5.4. 需求树及预分配

3.5.4.1. 需求树维护

从测试的维度，可以维护测试的功能项，项目的需求也在需求树上，可以通过拖拽的方式修改节点间的父子关系，以及双击修改和预分配人员。

测试需求项（功能模块目录），以一颗树的形式显示，双击修改，拖拽移动，也可同级上下移动调顺序，在用例管理，缺陷管理，接口用例中，都可以增加测试需求项，但是只有测试需求管理中可以拖和修改。

3.5.4.2. 预分配

预分配为可选流程，从预字也看得出，表示预先，提前的意思，其功能背景及目的如下：可以为非叶子节点预先设置开发，或是预先设置分配人员。如测试流程中，不含分配流程，不能预先设置分配人。该功能不常用，小项目，或小团队项目，不需要做此设置，只有大项目，大团队，开发人员很多时，可以为一级模块指定分配人员，或是开启分配流程时，为一级模块指定分配人员，这样，当测试人员提交 Bug 时，可在指定的人员范围内选择预定的人员。如某个节点指定了一个开发人员，当 Bug 流转到开发人员时，自

动指定该人员为下一处理人（在分散的团队中，为模块设置好开发人员或是分配人员很重要，这时测试和开发往往不在一起，测试有时不知道 Bug 流转 to 谁的节点下，大项目，就算转给他分配人，分配人也不一定清楚，应该分给哪个开发人员，但是项目在维护测试需求时，通过与 PM 沟通后，可以把沟通的结果，通过测试需求的人员预分配，设置到对应需求项上）。

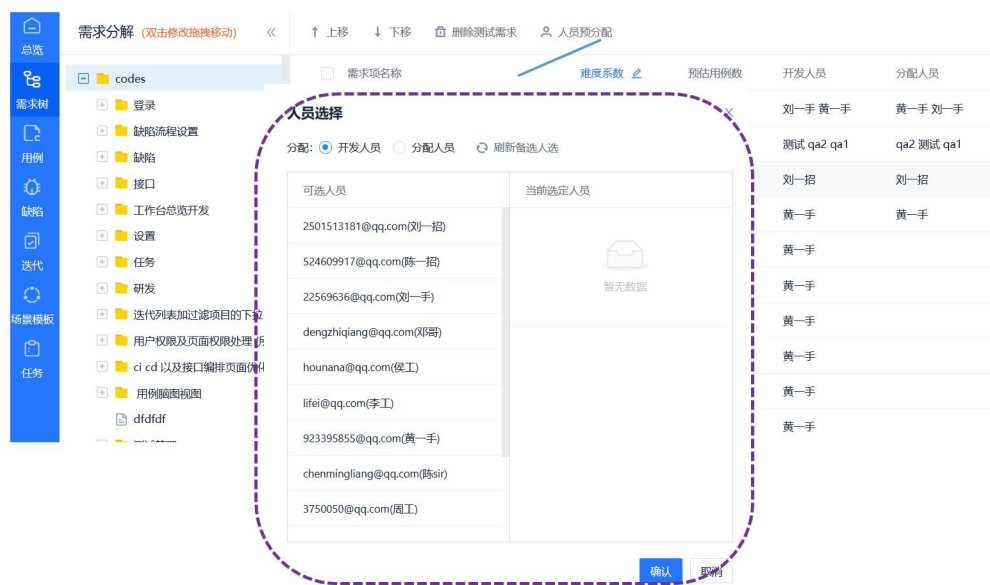


图 3-29

3.5.5. 脑图用例维护

Codes 中，脑图保持着脑图本真的使用习惯，在格式上没有任何限制，脑图转用例的规则是：叶子转为用例描述，非叶子作为功能模块路径。也可以以脑图视图的模式，把标准用例以脑图的形式展现。

3.5.5.1. 基本流程：

(1) 编写脑图文件-----> (2) 转为标准用例 -----> (3) 执行用例

以 Web 脑图形式维护用例，然后转标准用例。转标准用例后，只能在脑图中修改叶子，也可以在标准用例中修改描述，上面这两种修改都会双向同步；在脑图上或标准用例中执行用例，他们之间能自动同步

执行状态。有和 xmind 一样的快捷键，enter 增加兄弟节点，insert 或 tab 增加子节点，ctrl+v 复制当前节点(如含子节点连子节点一起递归复制)，ctrl+v ，粘贴复制的节点到当前节点下。如果是 mac 系统 ctrl 键更改为 win 键。

脑图用例转为标准用例后，还可以在脑图上操作增加删除修改，但是仅能点击右键菜单的方式进行操作，加入用例或是子节点以及修改节点及用例，也可以删除，只是一次只能删除一个节点且要从叶子开始才可以删除。

3.5.5.2. 脑图用例维护功能界面：

点击用例主界面右上更多下的“脑图用例维护”进入，脑图维护主界面如下图：



图 3-30

3.5.5.3. 注意事项：

转标准用例时，必须选择迭代，转标准用例后，可以直接点击右键菜单中执行。脑图用例维护也支持导入 xmind 导入，和标准用例的 xmind 导入的区别在于，这里导入，不会转为标准用例，只是以 WEB 展

示导入的脑图，并可修改。

3.5.6. 标准用例维护

有 4 个入口 1、工作台我创建的用例 TAB 下；、测试下用例中或定制的一级菜单用例中；3、用例库中；4、离线维护（以导入导出的形式）在同步到线上是同步不是导入 。标准用例是相对于脑图用例来说的。

创建测试用例

用例描述 *

填写描述内容

前置条件

填写前置条件

过程及数据 *

请填写过程及数据(潜质及预期都写在这里也可以。不重格式重内容)

期望结果 *

请填写期望结果(如过程及数据中填写了，这里写过程及数据)

备注

填写备注内容

需求 *

codes/登录

类别 *

请选择

优先级 *

请选择

标签

+

搜索标签

lalani

+ 创建标签

基线

请选择

成本 *

请填写 -单位5分钟

这三个下拉来源为字典中维护的数据

点击维护标签名,可现增加标签,也可从字典维护的标签中选择

确认

确认并继续

取消

图 3-31

3.5.7. 标准用例脑图视图

点击右键进入菜单，可以标准用例执行方式执行用例，也可查看用例详情，及以增加用例。



图 3-32

3.5.8. 用例导入导出(线上线下载步)

3.5.8.1. 概述

Codes 的导入导出，和传统的导入不一样，导出后可离线修改，执行和增加新用例后再同步(导入)到线上。例如从迭代 A 导出，要同步线上时，也要在迭代 A 中导入，否则无法通过同源校验。

图 3-33

3.5.8.2. 导入|同步

有 4 种导入方式：1 从 EXCEL 中导入；2 从 xmind 思维导图导入；3 从公共用例库中导入；4 从产品用例库中导入。从公共用例库导入时，如发现未维护用例库，会有相关提示，如请管理员维护用例库用例

类别。导入时，编写人员为当前执行导入的人员。[公共用例库与产品用例库的区别](#)

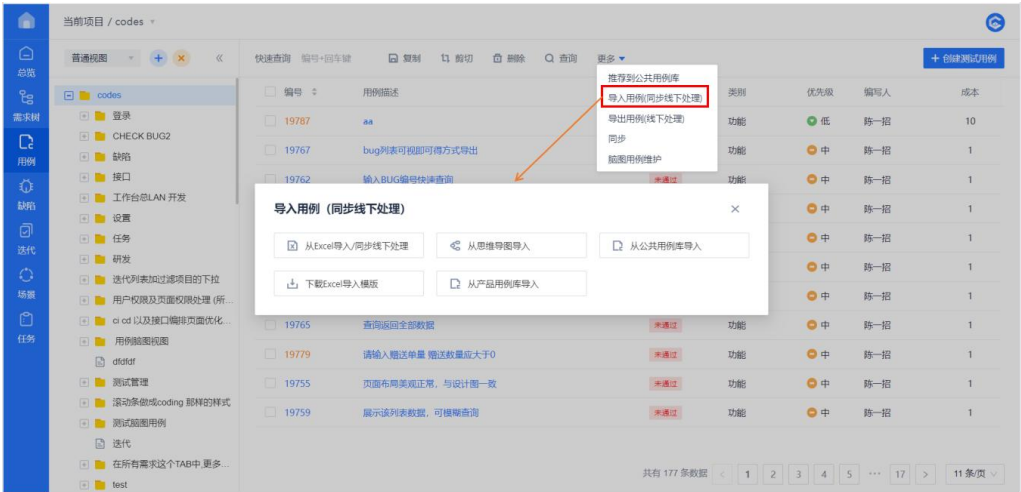


图 3-34

注意 2：从 Excel 导入时，如 Codes 系统内无 Excel 中的 优先级，类别，标签，以及功能项（也就是模块或需求）时，Codes 会自动创建。Excel 还可离线执行再同步导入，详见用例执行部分。导入格式见导入模板。

1	<div><div>codes</div><div>Codes 数智化研发效能平台</div></div>													
2	1、新增用例的导入，需将“编号/校验码”的值设置为“add”，导入后会在系统中新建不存在的模块													
3	2、新增用例的导入，功能项（在用例库中，为目录，规则一样）写不含项目名(根节点)的完整路径，模块间用中文斜杠(/)隔开；													
4	3、当第1，2列不修改时，导出的用例重新导入，会对原有用例进行更新或者修改													
5	4、第3行任何情况下不能删除；													
6	5、第4行的内容不能删改，它对新增用例的导入没有影响。													
7	6、用例库中导入时，将不处理线下执行的数据，只新增用例导入或是同步离线修改的内容													
8	不要删除本行，可导入前线下新增用例并执行，或只新增，如执行，可按照第6行设置离线执行及执行版本下拉列表，然后从人选取要执行的结果及版本，版本和项目绑定，线下执行前要按在线上项目中下载模板													
9	项目名称	codes 导入导出通用模板												
10	编号/校验码	功能项/目录	用例描述	前置条件	操作过程及数据	预期结果	线上状态	离线执行状态	执行版本	项目PM	刘一手	源校验码	无	
11		缺陷								类型	优先级	备注	标签（多个标签空格隔开）	编写人
12		缺陷/迭代/报告/调整												
13		缺陷1												
14		缺陷2												
15		缺陷3												
16		缺陷/用户中心开发												
17		缺陷/用户中心开发/设计												
18		缺陷/用户中心开发/项目维护												
19		缺陷/用户中心开发/分类模式开发												

图 3-35

从 Excel 导入时，需要下载 Codes 提供的模板，按模板进行导入，格式同上图。另外，类型和优先级，如果填写的不是基础字典维护里的数据，导入时会自动增加到基础字典中。**第 2 列为模块路径，规则如下：**

用斜杠隔开且不包含项目名，如：系统管理/用户管理，导入的时候，Codes 会自动检查，模块路径对应的测试需求项是否已存在，如果不存在则会自动创建，以斜杠来分级。

下载导入模板时，会导出当前项目的模块或需求目录到 Excel 中。

3.5.8.3. 导出|离线处理

以所见即所得方式导出用例，也就是，按当前查询条件导出用例。

注意：导出的数据中第一列为编号/校验码，用于离线处理再同步回 Codes 中，源校验码也用于同步线下处理时确认项目。详见导出文件中的说明。

3.5.8.4. 产品用例库同步用例

项目中，从产品用例库，或是公共用例库中导入了用例，后续用例库中用例做了修改，通过同步，可以把所做修改同步过来，如发生变动同步会生成同步变更历史。还会把新增的同步过来。支持双向同步。在哪里点击同步就是以哪里为目的地，然后选择要同步的源。

3.5.9. 用例执行

从编写用例到执行用例流程：录入需求或直接维护模目录---->编写用例---->把用例加到迭代下--->执行用例；也可以把单个用例组成场景，分配场景到迭代下执行。

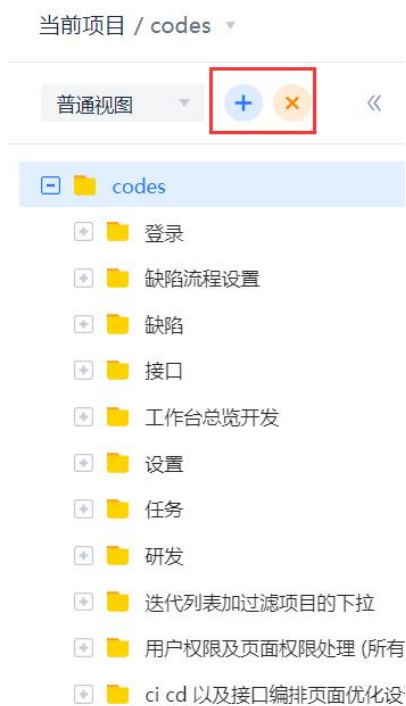


图 3-36

3.5.9.1. 离线执行/处理用例

从迭代下的测试用例或是测试场景中，用例中导出到线下，然后在线下增加，修改，执行以及这三个操作的排列组合处理后，再导入同步到线上 Codes 中。



图 3-37

3.5.9.2. 在线执行/处理用例

有 3 个入口：1、工作台我的待办下或全局事项下；2、工作台迭代计划从迭代中用例或场景中；3、脑图用例中；

点击用例状态，可执行用例。

如不在迭代下，从用例维护中点用例状态执行用例时，必须要选迭代才能提交。

3.5.10.用例同步

项目中，从产品用例库，或是公用用例库中导入了用例，后续用例库中用例做了修改，通过同步，可以把所做修改同步过来，如发生变动同步会生成同步变更历史。没有变动则什么也不变。还会把新增的同步过来。

可在项目以及产品用例库间“双向”同步用例。所谓的“双向”同步，指用例在产品用例库中修改了，可以同步到项目中；项目中的用例修改了，可以同步到产品用例库。

项目 1 的用例导入到产品库 1 可以做到：

- 1、项目 1 中用例修改后，可以在产品库中同步
- 2、产品库中用例修改后，可以在项目 1 中同步

此时如果产品库中用例导入到项目 2，也可以做到：

- 1、项目 2 中用例修改后，可以在产品库中同步
- 2、产品库中用例修改后，可以在项目 2 中同步

3.6. 接口测试

3.6.1. 概述：

Codes 接口测试采用无代码方式，对测试人员友好，资深玩家也可以编码实现接口测试。Codes 包含 6 大功能：1、接口设计(接口文档和 Mock)，2、场景配置，3、接口用例，4、接口扩展（接口插件、SQL 接口、录制回放），5、场景编排，6、定时执行和 webhook 执行。

[接口测试](#) 要切换到经典布局才能看到菜单

3.6.2. 基本流程：

baseUrl 设置-----> 基础认证设置 ----->维护接口用例----->创建接口测试场景(可在迭代中直接增加)--->手动执行接口测试场景(可单个，也可一键执行场景中所有接口)或者定时执行测试场景。

3.6.3. 接口测试主界面：

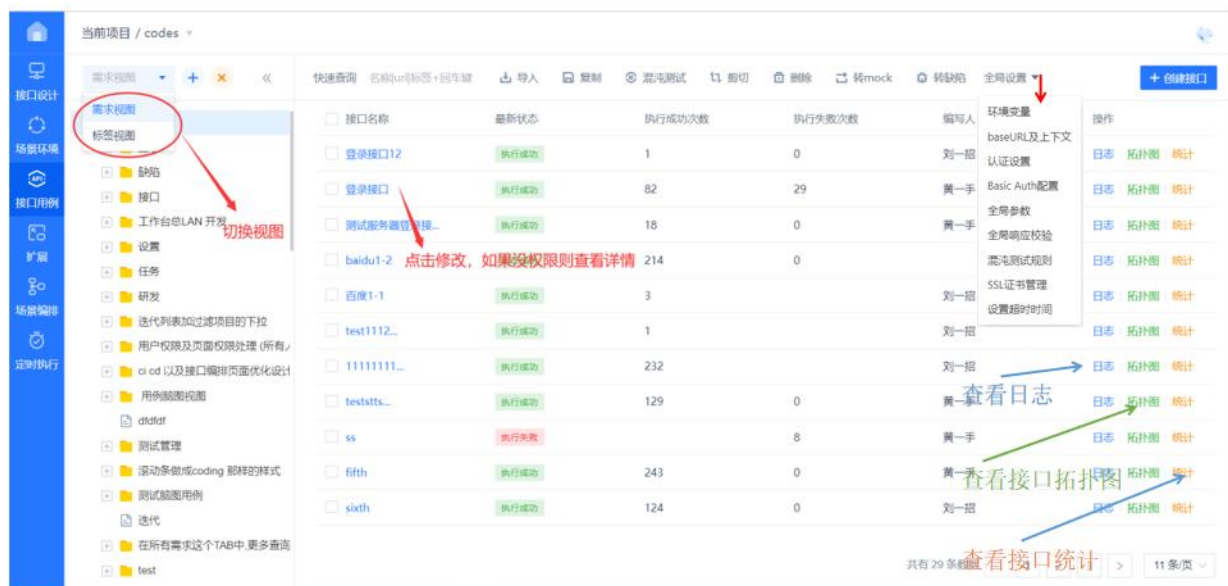


图 3-38

3.6.4. 全局设置

切换项目同手工用例测试一样。在左侧的接口用例管理中，设置全局参数。**如果看不到全局设置，需要用“管理员”登录并授权。**

3.6.5. BaseUrl,环境变量，绝对 URL 优先级关系

如有环境变量，baseUrl 失效，如有绝对 URL 也就是以 http 开头完整接口的 URL，那环境变量，baseUrl 都失效。下图中，红框内以 http 开头，就认为是绝对 URL。



图 3-39

3.6.6. 接口导入

支持从 postman、JMX、swagger 、httpRunner 2 导入。postman 导入时，目录和 postman 中的一致；swagger 导入时，会按参数类型设置好缺省值。

从 swagger 导入时,同时会生成接口文档,且在接口用例中,有一个查看文档的链接;另外 swagger 导入时，会导到当前节点下，且 tags 会作为目录，加到当前节点下。

3.6.7. 接口加密，解密，签名，前置处理，后置处理

在接口->扩展->接口插件中上传你的：加密、签名、解密、采样器，前置处理、后置处理的 JAVA 源文件或是 class 的 jar 包。在接口用例页面，可选择所需要加密、签名、解密算法、前置处理、后置处理，执行接口时，就按选择的加密、签名、解密、前置、后置进行处理。



图 3-40

3.6.8. 接口参数化

3.6.8.1. 内置参数化

Codes 中没有全局变量，要是想使用，首先需要创建一个 Mock 接口，从 Mock 接口中提取全局变量，在其它接口中引用即可。

用于测试时，需要每次执行设置不同的参数值，语法及规则如下，注意语法是区分大小写的，必须与下面列出来的一致

%(Sdate) 表示 YYYY-MM-DD 的日期

%(int) 表示自增，用一次加 1,每个项目一个自增系列

%(Ldate) 表示用的是执行时，系统时间含时分秒 YYYY-MMM-DD hh:mm:ss

%(timeTs) 表示用的是执行时，系统时间的 13 位时间戳 如, 1609913571818 ,可以加减一个你输入的分钟数

%(timeTs-5) 表示当前系统时间戳减 5 分钟， %(timeTs+5) 表示当前系统时间戳加 5 分钟

%(value) 只要括号中不是 Sdate ,int ,Ldate, timeTs 表示是括号中的值加一个随机数来参数化。

邮箱、UUID 、IP 、域名、手机号、身份证号、流水号（可以设置位数）、mac，这些后续也作为内置参数化实现。如下图所示：



图 3-41

3.6.8.2. 内置函数

除上面提到的 4 个自定义的参数化外，还有 41 个内置的常用函数，例如：加解密函数、四则运算函数、常用函数等。在接口用例维护页面中，点击带问号的图标显示函数帮助，函数间可以相互嵌套使用。41 个内置函数（常用函数、加密函数、随机函数、四则运算函数），内置函数在 head 和 body 中都可以使用，如下图所示：

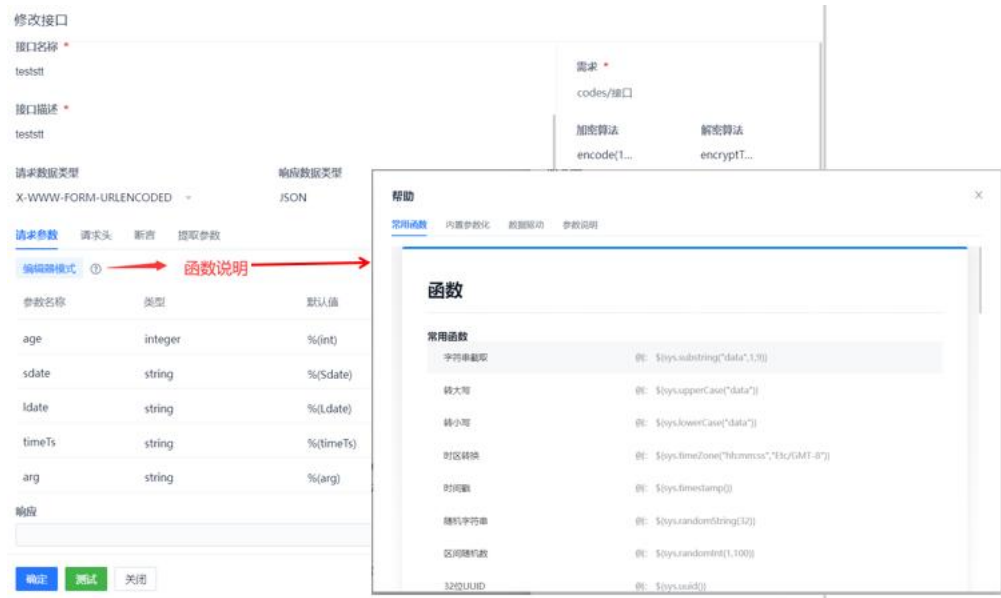


图 3-42

3.6.8.3. 用户自定义采样器



图 3-43

3.6.9. 接口间参数传寄

例如：接口 A 的参数值要引用另一接口 B 的响应的输出时，需要 B 接口中提取参数，然后在 A 接口中选择或是手动输入。可以在接口的请求头中引用参数，也可是接口的请求参数中引用其它接口提取的变量，注意全局请求参数中，不可以引用其它接口提取的变量，这一限制不是技术原因，主要了防止滥用，从而引起接口循环依赖。

注意：如果提取的参数是布尔或是数值型时，可以在参数名后加上_nostr，在引用该变量处理 json 时则不会加上引号。

3.6.10. 拖拽式接口参数提取

指从接口响应数据中提取参数供其它接口使用。

新增接口时，需要先保存接口，保存时（如是修改接口，之前没有返回响应数据，需要先“测试”），

Codes 会自动执行一次测试，得到响应的数据结构且是 json 时，才可以手动输入，或是点击“提取参数”tab 下的，拖拽提取参数页面，拖动需要的提取的属性，设置好变量名，可以提取多个参数。如下图所示：



图 3-44



图 3-45

3.6.11. 引用提取的参数值

指接口 A 引用接口 B 提取的参数。引用参数语法为：\$(变量名形式)，**head 中也可以引用参数。**

在键值对模式时，可以从下拉中选或是手动输入。



图 3-46

在键值对模式和编辑器模式都可以手动输入



图 3-47

3.6.12.编码接口参数提取

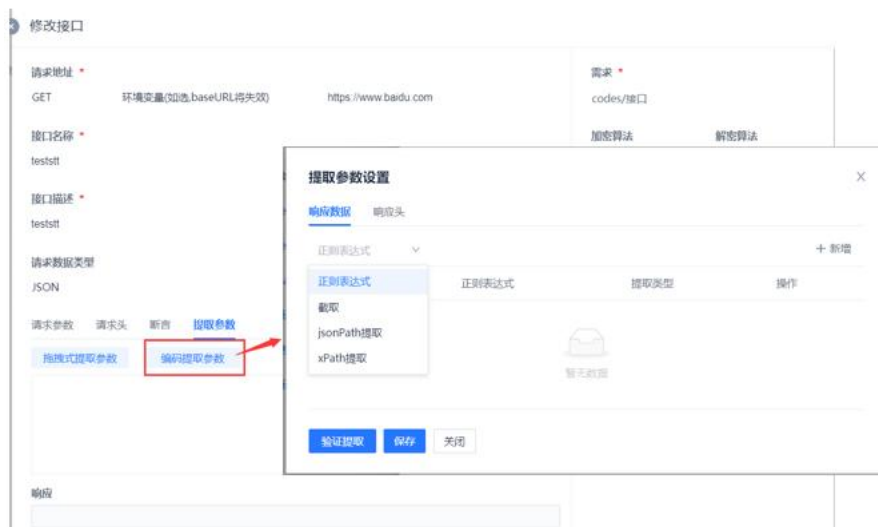


图 3-48

3.6.13.接口断言

断言主要用于校验接口的返回是否符合预期，如接口没有指定断言，就采用全局断言，如果全局断言也没有，就用 http 响应码，http 响应码 200 和 304 都认为通过。在接口的响应中，如断言不通过，接口响应数据的背景显为红色，通过是绿色，如果接口没设置断言，全局的也没有，显示为紫色。

3.6.14.拖拽生成断言

新增接口时，需要先保存接口，保存时（如是修改接口，之前没有返回响应数据，需要先“测试”），Codes 会自动执行一次测试，得到响应的数据结构且是 Json 时，才可以手动输入，或是点击拖拽式”维护断言”tab 下的，在拖拽维护断言页面，拖拽要校验的一个或多个属性，选择操作符，设置好值。如下图所示：



图 3-49



图 3-50



图 3-51

3.6.15.编码生成断言

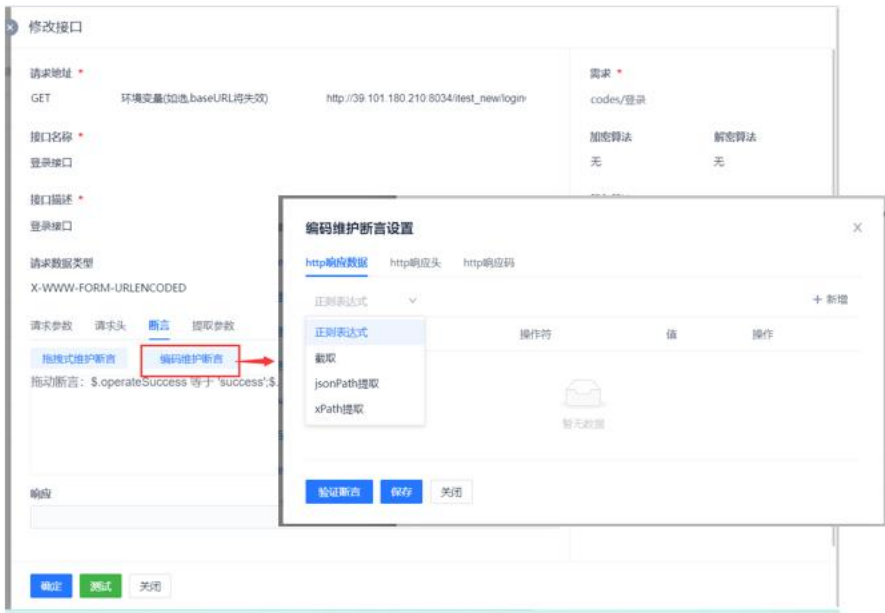


图 3-52

可对响应数据，响应头，及响应码进行断言

3.6.16.特别提醒

3.6.16.1. 复杂键值对转 json

如果是 post 请求，不想写 json 格式的数据，可以写 user.loginname =XXX

user.password=XXX 这种形式，后台在调接口时自动转化。比如下图中的 key value：



图 3-53

后台调用接口时自动转为如下 json 格式：

```
{
  "user": {
    "loginname": "liuxingjian@qq.com",
    "password": "1234567"
  }
}
```

图 3-54

3.6.16.2. 美化 json

只有输入的是 json 时才能美化，如下图的原始输入：

请求参数 请求头 断言 提取参数

键值对模式 json美化 转键值对: ☒ ?

```
{ "user": {
  "loginname": "liuxingjian@qq.com",
  "password": "1234567" }}
```

图 3-55

美化后为:

请求参数 请求头 断言 提取参数

键值对模式 json美化 转键值对: ☒ ?

```
{
  "user": {
    "loginname": "liuxingjian@qq.com",
    "password": "1234567"
  }
}
```

图 3-56

3.6.16.3. 接口前后置 SQL 处理

Codes 中可以通过 [API 生成](#)来处理: Codes 中如果你需要到数据中验证,你可以通过 Codes 的 API 生成功能。创建好数据源后,写 SQL 就行,SQL 转变为服务(接口)。当你需要进行接口测试时,编写一个接口用例,然后不管是执行完要验证数据,还是执行前,从 DB 中提取数据可以,引用这个接口用例的数据即可。

3.6.16.4. URLENCODED Body 设置说明

请求数据类型为: X-WWW-FORM-URLENCODED 时,有个 body 参数是否的选项,缺省为否。为是时表示,如 body 中有参数,请求的时候 Body 的数据不变,原样保留;如选否,则把 body 的参数追加到 URL 后。

3.6.16.5. 编辑器模式说明

编辑器模式下如下图所示,缺省都是转,如有非正常格式的数组时,选不转,这时,KV 和编辑器里的数据是两份,不只有一份,且点键值对模式时,不会把编辑器模下的内容转到 KV 中,反之亦然。



图 3-57

3.6.17.接口依赖拓扑

接口间只要存在参数引用,如 A 接口引用 B 接口输出的参数,A 对 B 就有依赖关系。

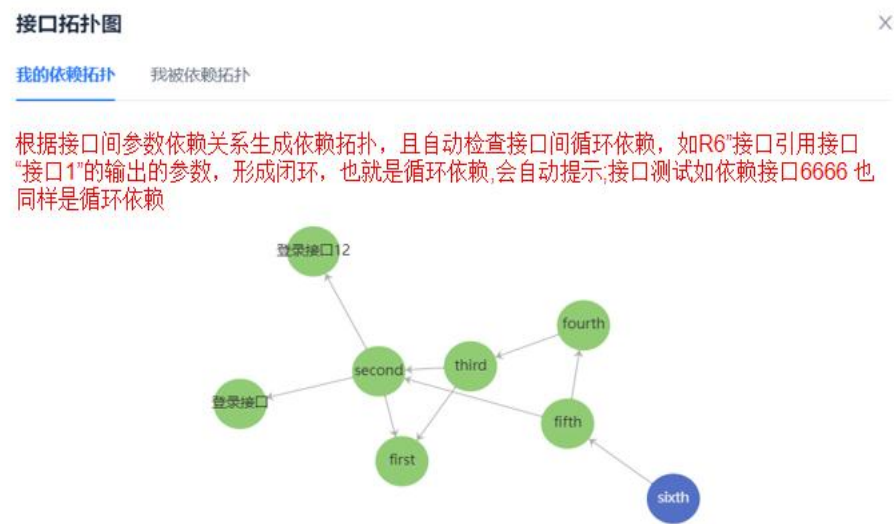


图 3-58

3.6.18.场景编排

3.6.18.1. 概述

作用同测试用例包，用于分配接口测试用例。使用方式同测试用例包。执行环境下拉可选，[参见接口环境](#)。



图 3-59

3.6.18.2. 手动执行接口测试用例



图 3-60

3.6.18.3. 数据驱动

场景支持数据驱动，CSV 中第一行的各列为接口中的参数名，Codes 按参数名，去换接口中对应的参数值。数据驱动，也是按面向对象的方式，方便复杂 json 的结构，传统的数据驱动，只方便 KV 方式，复杂对象，表达起来费劲，我们依然采用 xxx.xx.xx 这种对象属性访问形式。依然采用 xxx.xx.xxx 这种对象属性访问形式，即支持简单 KV ，又能一行表示一个 json 对象。

1	dto.user.name	dto.command	dto.isAjax	dto.user.id	
2	name1	add	ture		1
3	name2	add	ture		2
4	name1_1	update	ture		1
5	name1_2	update	ture		2
6		delete	ture		2
7					
8					
9					

依然采用xxx.xx.xxx 这种对像属性访问形式，即支持简单KV ，又能一行表示一个json 对像，直观又易于理解

图 3-61

数据驱动参数匹配规则：有三种

- 1. 匹配参数名；

- 2. 匹配变量名;
- 3. 参数和变量名同时匹配

xxx.xx.xxx 为匹配参数名, 如 dto.command, 匹配的是接口中参名为 dto.command 的参数, 当然也可以是 kv 形式中的 k 的名称, 如 name , age 等, dto.command 这表达复杂对象时的写法。参加数
名中的 KV 的中 K 如下图所示:

请求数据类型
JSON

响应数据类型
JSON

请求参数

请求头

断言

提取参数

编辑器模式 ?

+ 新增

参数名称	类型	默认值	操作
age	string	%(int)	删除
sdate	string	%(Sdate)	删除
ldate	string	%(Ldate)	删除
timeTs	string	%(timeTs)	删除
arg	string	%(arg)	删除
num	string	&(samplerTest)	删除
accessToken	string	\$(token)	删除

图 3-62

第 2 种按参数名, 也就是占位符来匹配, 也就是 KV 中 V 的名字来匹配,格式是: \$(变量名),这时在 csv
中第一列就要就要写这种格式, 如下图所示:

请求数据类型		响应数据类型	
JSON		JSON	
请求参数	请求头	断言	提取参数
编辑器模式	+ 新增		
参数名称	类型	默认值	操作
age	string	%(int)	删除
sdate	string	%(Sdate)	删除
ldate	string	%(Ldate)	删除
timeTs	string	%(timeTs)	删除
arg	string	%(arg)	删除
num	string	&(samplerTest)	删除
accessToken	string	\$(token)	删除

图 3-63

第 3 种 accessToken:\$(token), 这是双匹配模式, 要求, 参数名, 也就是 KV 中的 K 要是 accessToken 同时 v 中的变量要是\$(token)

这三种匹配模式可以混合使用, 如 CSV 中, 不同列可以用不同的匹配模式, 推荐使用第二种。

3.1.17.2 场景中用分配接口模式说明

如下图所示, 分为手动和自动模式, 这里的手动, 自动指用例在场景中的执行顺序排序。

自动时: Codes 会自动加入没选择的, 但被已选择的接口用例依赖的接口用例, 同时会按依赖关系自动排执行顺序, 自动排序后还可以进行手动调整。

手动时, 按选择的顺序, 作为执行的顺序, 如选的接口用例中, 依赖了别的接口, 但是没选被依赖的用例, 会给相关提示。

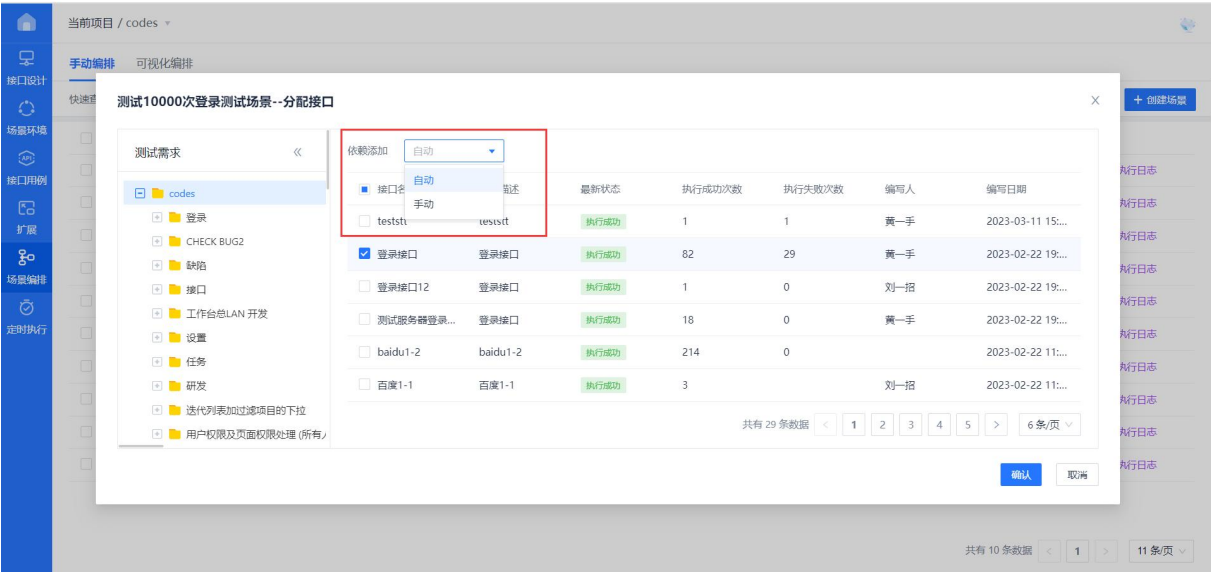


图 3-64

3.6.19.接口 Mock

在接口菜单下，左则二级菜单接口设计中，维护好 Mock 接口的 URL、请求参数、请求头、响应码、响应数据，发布后即可调用 Mock 接口，访问 URL 为：安装 Codes 服务器的 IP:8064/{你 Mock 接口中维护的 URL}。另外请求 cookie 以及响应 head，响应 cookie 设置和 4 种响应延时设置，以模拟响应延迟及网络不好的场景。如果设置 contentType 在 Mock 响应头中加 Content-type 值等于 application/json 等。

注意接口地址是相对地址，不能填完整的，Mock 的访问地址为 Codes 的 ip 地址，端口为：8064 如下图所示：Codes 安装服务器 IP 为 192.168.1.168 则访问下图 Mock 地址为 <http://192.168.1.168:8064/list>

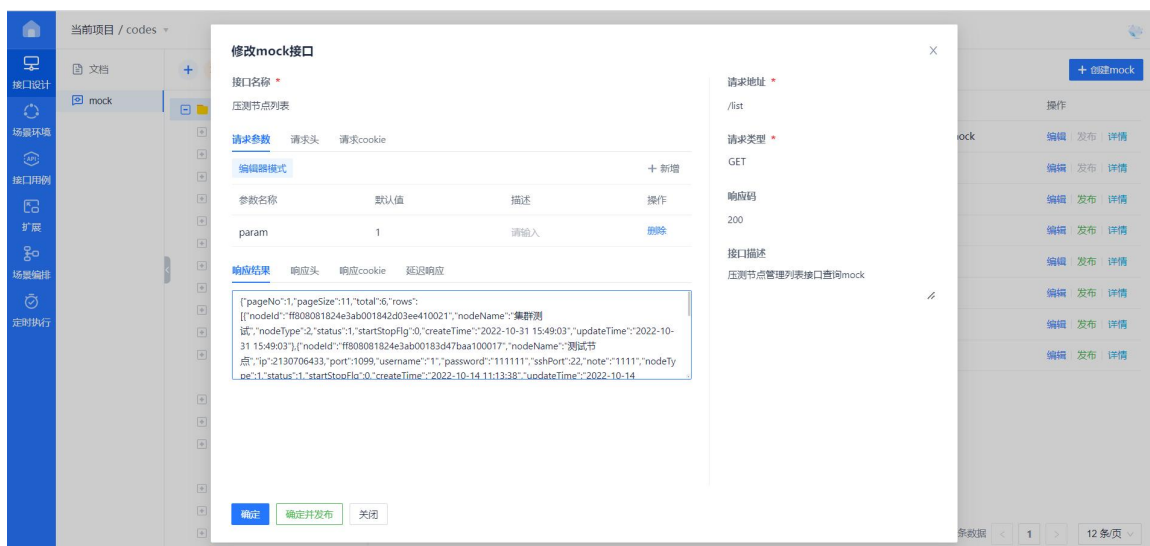


图 3-65

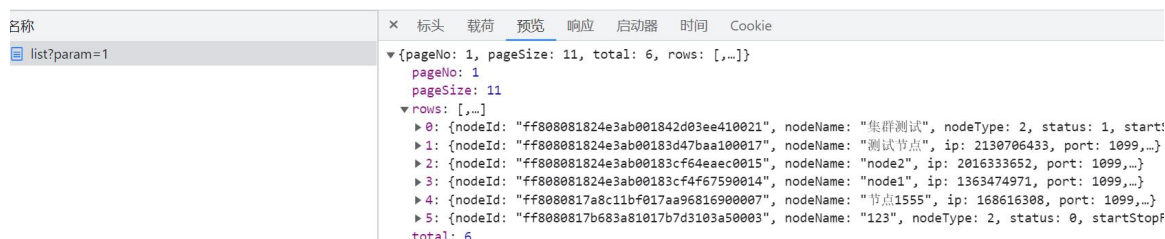


图 3-66

3.6.20.接口插件维护

在接口用例管理中可选择的加密、解密、签名、自定义采样器、前置处理、后置处理算法，其来源于维护数据，最终实现接口测试代码和加密、解密、签名、自定义采样器、前置处理、后置处理算法分离。

可以上传 jar, jar 里是编译好的 class 文件, 如果加密、解密、签名实现依赖第三方包, 可以使用此方法: 把实现类的 class 和第三方包的 class 合为同一个 jar (如有多个 jar 请合并为一个)。

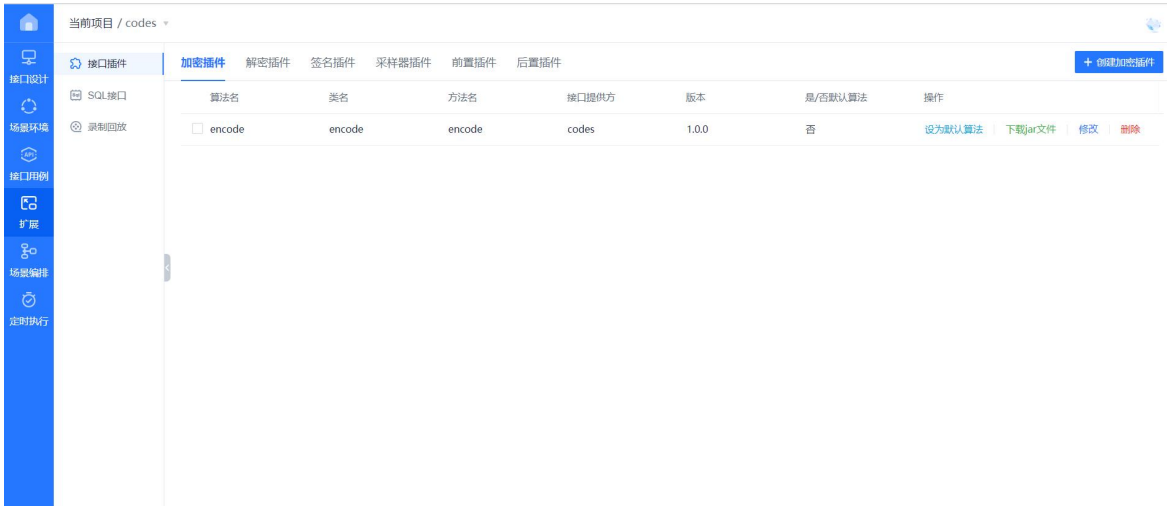


图 3-67



图 3-68

注意事项 1：类名要填含包名的完整路径类名，方法名任意合法方法名，且只能有一个 string 类型的参数，返回 string，**加密和解密的方法定义如下：**`public String 方法名 (String str) {}`；

注意事项 2：签名类名和方法名和加解密一样，但是方法的定义不一样，入参是个 `TreeMap<String,String>`，返回是个 `HashMap<String,String>`，Codes 遍历返回的 map，把值写入 http head 里，以 map 中的 key 为 head 里的参数名，**签名方法定义如下：**`public HashMap<String,String>`

方法名(TreeMap<String,String> paraValues){} ;

签名方法中，传入的 TreeMap<String,String> ，是所有请求有参数，key 为参数名，value 为参数值，有些签名是带防篡改的，需要请求的参数及值，所以传入，是否使用由实现者自己的需求来决定。

采样器：类名要填含包名的完整路径类名，方法名任意合法方法名，无参数，返回 string，**采样器的方法定义如下：public String 方法名(){} ；**

前置处理：类名要填含包名的完整路径类名，方法名任意合法方法名，且只能有一个 string 类型的参数，返回 PluginResDataVo，**前置处理的方法定义如下：public PluginResDataVo 方法名(String str){} ；**

```
1 package cn.com.mypm.apiManager.conts;
2
3 public class PluginResDataVo {
4     private String dataType;
5     private String data;
6
7     public String getDataType() {
8         return dataType;
9     }
10
11     public void setDataType(String dataType) {
12         this.dataType = dataType;
13     }
14     public String getData() {
15         return data;
16     }
17     public void setData(String data) {
18         this.data = data;
19     }
20 }
21
22
```

图 3-69

PluginResDataVo 类中，有两个属性，String data：返回数据，String dataType：返回类型；dataType 的返回类型有 **TEXT**，**JSON**，**XML**，**ROW** 实现者根据自己具体返回设置相应类型；提供的 jar 包中提供了 **ResultConstants** 抽象类，也可以使用。

```
public abstract class ResultConstant {  
    public final static String TEXT = "TEXT";  
    public final static String JSON = "JSON";  
    public final static String XML = "XML";  
    public final static String ROW = "ROW";  
}
```

图 3-70

后置处理：类名要填含包名的完整路径类名，方法名为任意合法方法名，且只能有一个 string 类型的参数，返回 string，**后置处理的方法定义如下：** `public String 方法名(String str){}`；

限制说明：如果你上传的是 jar，class 需兼容 jdk 1.8 编译。

加密、解密、签名、采样器、前置处理、后置处理的类必须包含无参构造函数。

3.6.21.接口定时执行

以定时任务的形式，自动执行测试场景下的用例，可以选择多个测试场景。

3.6.22.接口场景环境

主要用于在接口测试场景中切换不同的测试环境。其中 baseUrl 为必须填写项，认证配置、全局参数、超时时间为非必须填写项。（如非必须填写项未填写时，则引用原接口中的配置）在接口测试场景中，选择特定场景后，所选择环境中下述 4 个表的配置项，都应用于场景内的中所有接口配置的数据。



圖 3-71

3.6.23.接口调用链日志

在接口测试场景列表中，点击执行日志，显示一个列表，点击查看日志，显示调用链。如下图所示：

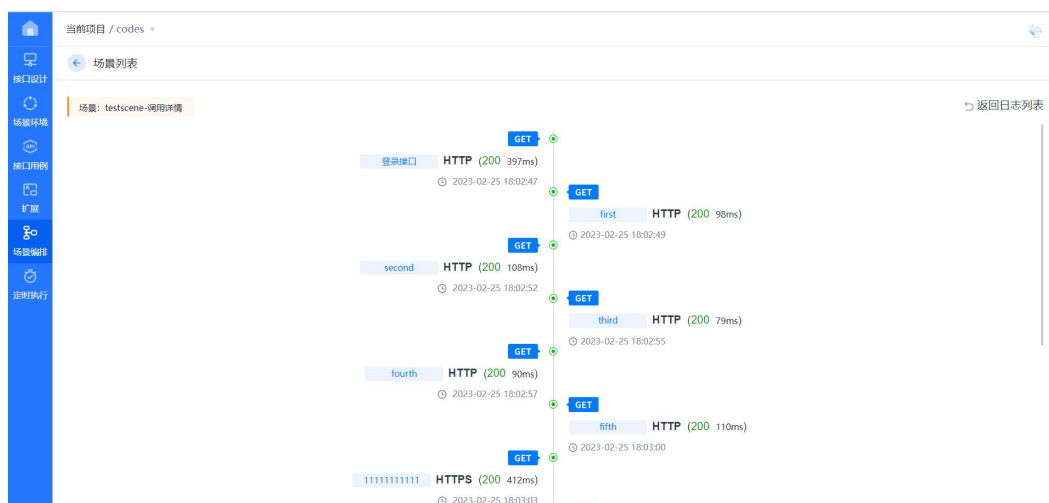


图 3-72

3.6.24.全局变量使用

postman 中全局变量，是通过直接定义变量来实现；Codes 中全局变量使用，稍有不同，在 Codes 中，创建一个 Mock 接口，返回你要返回的数据，然后再创建一个接口，调用该 Mock 的 URL，并提取参

数，其它接口则引用这个参数，能使全局变量服务化，不同的环境配置不同的 Mock 即可。

3.6.25. 接口文档

Codes 中可以手动维护接口文档，且可发布为 Mock；另外也可以从 swagger 导入(同步)接口用例时自动生成接口文档，但注意：swagger 生成的接口文档，只能查看不能修改，只能在 swagger 中修改后再同步。在 Codes 上提交的接口文档发布时，会自动生成 Mock 以及接口用例，且会自动加入到 Mock 服务，以及 Mock 接口的两个目录中，如下图所示，从 swagger 同步过来的接口文档发布 Mock 时，则不会生成接口用例，以及生成这两个目录。

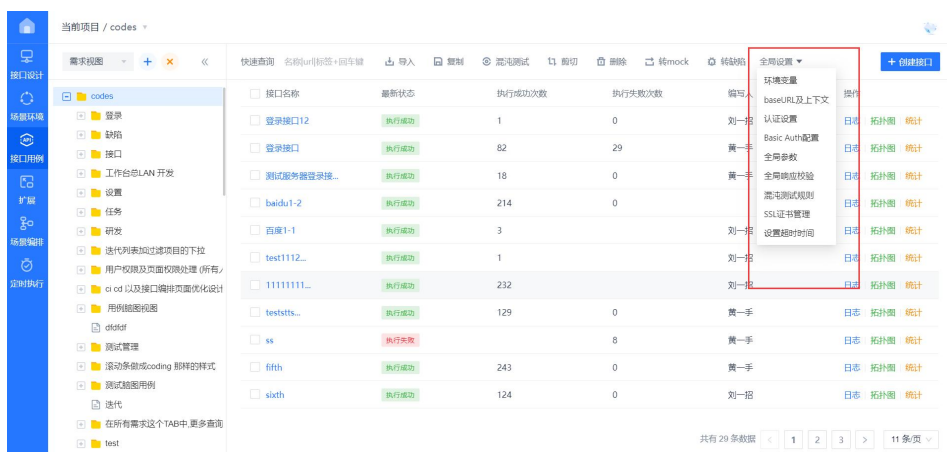


图 3-73

3.6.26. CI/CD 等外具工具触发执行接口测试

Codes 有对外接口,比如在jenkins 中 build 完成并部署后,在job 中或量 beanshell 中 通过 curl 执行，Codes 上的定时任务的测试场景，在配置 curl 时，需要带上在 Codes 上配置的安全访问 key 。除了配置访问的 key 也就是 token 外，Url 中还要带上要实时执行的 Codes 上的定时任务的编号，可以是多个用逗号分开，或是传 taskId 则执行所有测试场景。Url 如下所示：

http://codes 服务器 IP:端口 /codesBackend/jenkinsCallBackCodes?token=XXXX&ids=xxx,xxx,xx 或 http://codes 服务器 IP:端口 /codes/jenkinsCallBackcodes?token=XXXX&taskId=xxx

token 为系统管理下,安全访问 key 中配置的启用状态的jenkins key ;传 taskId 时跑所传项目 id 下的该项目的接口自动化定时任务,传 ids 时,跑指定的 1 个或多个定时任务,多个定时任务编号用逗号隔开。

4、设置

4.1 用户管理

租户在 Codes 官网注册后,会生成一个默认的管理员账号。系统激活成功后,**请特别注意管理员首先需要:以管理员登录 Codes,通过“个人信息维护”修改管理员默认密码以及 e-mail。**



图 4-1

4.1.1 邀请用户

如下图所示,输入需要邀请人员的邮箱地址,被邀请人员收到邀请邮件,点击确认,即表示邀请成功,列表上可看到邀请成功的人员列表。

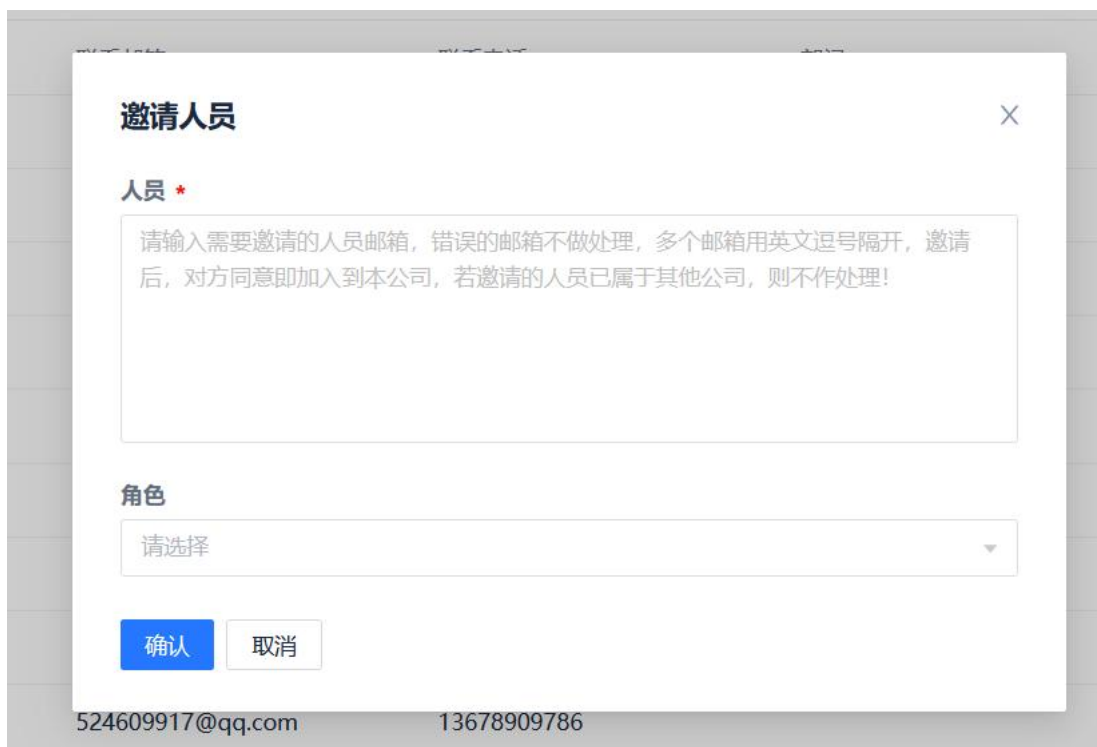


图 4-2

4.1.2 修改用户

修改用户基本信息。

4.1.3 设置用户是否可查看所有项目

设置用户是否可以查看所有项目。也就是设置为管理员。这里只是数据权限，管理员的功能权限也是从角色获取的。

4.1.4 密码重置

重置后会随机生成密码发送到被重置用户的邮箱。

4.1.5 删除用户

将用户从该租户中移除。

4.2 用户组管理

一个用户可从属于多个组，一个组可含多个用户；组不具有权限，只是便于用户的查找群用户的一个集合。组与用户的关系，可双向维护（即可以在用户维护中指定其所属组，也可以在组维护中指定其成员）。即可在增加用户组时，从下拉列表中选组员，也可在用户组列表中，通过人员维护的方式维护选择组员（人员多时推荐这种方式）。

4.3 权限管理

4.3.1 角色管理概述

Codes 的权限细化到“按钮”级，权限授予到角色对象上，用户通过从属于角色来获得所从属的角色

的权限，例如：用户从属于多个角色，其权限为多个角色权限的并集。

[illegible]

图 4-3

4.3.2 角色人员维护

在角色列表中，点击某个角色操作列上的人员维护，可增加或删除角色内人员。

人员维护账户列表

角色内账户

角色外账户

邮箱+回车键

从角色中删除

<input type="checkbox"/> 姓名	联系邮箱	联系电话	员工编号	职位	状态
<input type="checkbox"/> 周工	3750050@qq.com	13678913678			启用
<input type="checkbox"/> 黄一手	923395855@qq.com	16572186666			启用
<input type="checkbox"/> 刘一招	2501513181@qq.c...	15202845712			启用
<input type="checkbox"/> 陈一招	524609917@qq.com	13678909786			启用

共有 4 条数据

< 1 >

7 条/页

图 4-4

4.3.3 角色权限维护

在角色列表中，选中某个角色，点击“权限维护”，可修改或是增加角色权限。

4.4 基础字典维护

4.4.1 概述

维护下拉列表数据，以及设置缺省下拉选项，当前只有缺陷管理类，缺省下拉选项有效。数据项为蓝色的表示系统缺省的，缺省的不可以删除，只可以停用，只有自定义的测试基础数据可以被删除。当前可维护下面 14 类基础字典数据。

4.4.2 测试用例类

用例优先级和用例类型。

4.4.3 缺陷管理类

缺陷频率、缺陷类型、缺陷等级、缺陷时机、缺陷优先级、缺陷来源、缺陷引入原因、缺陷引入阶段、缺陷发生平台。**可以设置缺省项，设置为缺省项后，在新增缺陷时，这些字典默认选择设置为缺省项的值。**

4.4.4 任务管理类

任务类别，任务紧急程度，任务难易程度。

4.4.5 用例标签

有用例公共标签，以及接口用例公共标签两种。

4.5 测试用例库

点击用例库中，某个用例描述，查看用例详情。

4.5.1 概述

公共用例库保存了通用用例，然后在测试项目中可以导入用例库中的用例，用例库来至测试项目中的用例，经推荐后，被管理人员审核通过后，正式加入到用例库，是一个全新的副本，不因项目中用例删除和修改而受影响。

所有人都可以浏览用例库，但只有管理人员才能对用例库类别进行维护和用例入库审核。

4.5.2 公共用例入库

选择要入库存的用例，点“推荐”，输入推荐理由，然后在用例库中，由管理人员审核后正式入库。

注意：用例入库时，不含用例的附件信息。

4.5.3 产品用例库

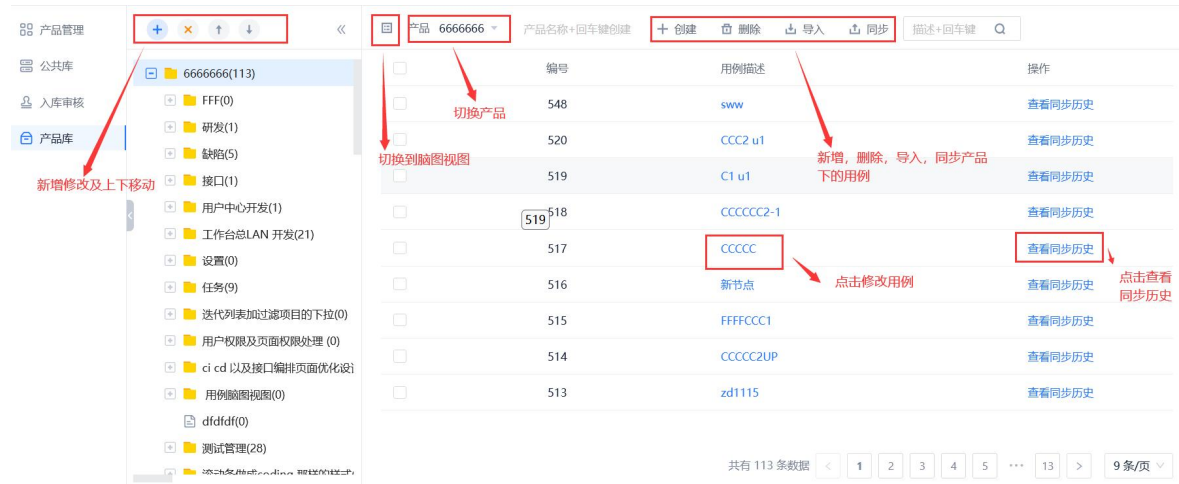


图 4-5

当产品库从项目中导用例时，如果当前在产品用例库中，选择了一个非根节点，那么导入的项目的模块目录，将挂到所选的非根节点下， 否则目录导入到根节点下。

产品用例库可以从项目中同步用例，同步在项目中修改过的用例，或是新增的。

4.5.4 产品管理



图 4-6

4.5.5 入库审核

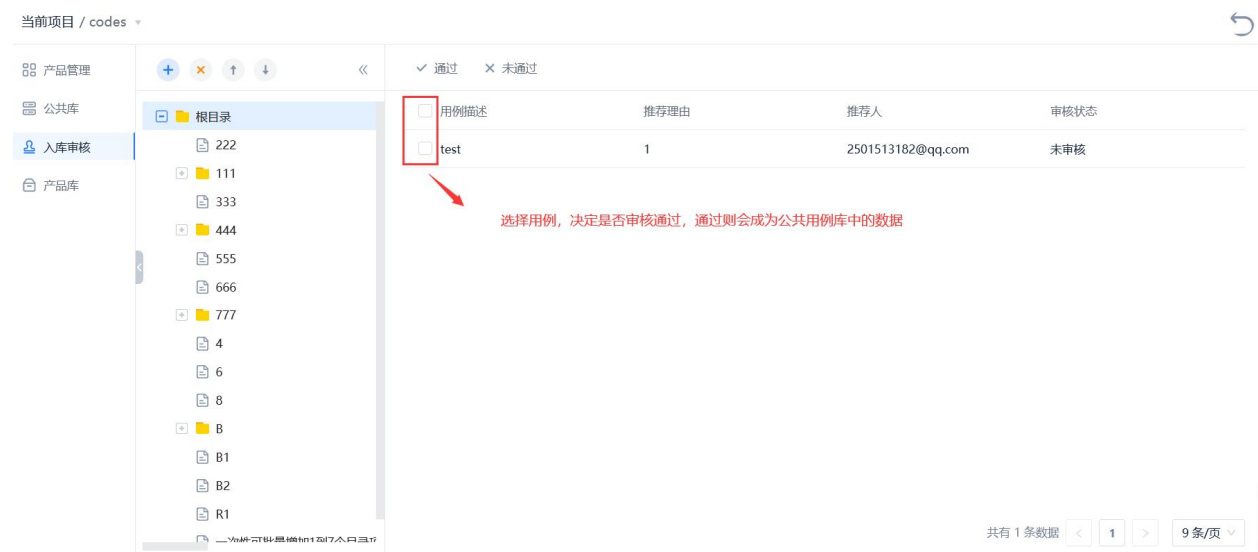


图 4-7

4.6 项目维护

项目有两类，一类是全流程也就是从需求到测试的完整项目；另一种纯管理就是指只有需求和任务管理的项目。且随时可以切换项目类型，切换到纯管理项目时，测试相关的功能菜单不显示。

3.5.0 之后的版本增加了项目模板，不再支持切换项目类型，详见如下功能说明

<https://icodes.work/blogPage/seventeen> Codes 终于迎来了 3.5.0 的重磅发布，3.5.0 实现了功能深度与易用性的平衡

维护项目基本信息，维护测试流，维护项目版本，创建项目入口可以在切项目的最下面点创建项目，也可以设置下的项目维护中。创建项目时，可以指定阶段模板，以生成里程碑相关的数据，且后续在工作台仪表盘 PM 可以再手工维护里程碑信息。在设置菜单下阶段模板中可以维护里程碑相关数据。

当前项目 / codes

快速查询

名称[编号|部门+回车键]

项目状态 所有状态

按项目状态过滤

+ 创建项目

<input type="checkbox"/> 项目名称	切换到分类视图	编号	部门	开始日期	结束日期	状态	操作
<input type="checkbox"/> jmeter 集成		test01	test01	2023-02-13	2023-02-14	进行中	<div>维护删除</div>
<input type="checkbox"/> CRM 对接		cd	sdsd	2022-12-21	2022-12-27	进行中	<div>维护删除</div>
<input type="checkbox"/> ci cd 重构		itest	itest	2022-12-14	2023-02-28	进行中	<div>维护删除</div>
<input type="checkbox"/> testLink 数据迁移		007	dev1	2022-11-07	2022-11-30	进行中	<div>维护删除</div>
<input type="checkbox"/> itest 数据迁移codes		21	21	2023-02-01	2023-02-10	进行中	<div>维护删除</div>
<input type="checkbox"/> chatGpt集成		222	321312	2022-12-22	2022-12-27	进行中	<div>维护删除</div>
<input type="checkbox"/> codes		1	测试	2022-11-09	2022-11-25	进行中	<div>维护删除</div>
<input type="checkbox"/> 测试		5	测试	2023-03-01	2023-03-28	进行中	<div>维护删除</div>
<input type="checkbox"/> 测试项目		4	dev1	2023-03-13	2023-03-30	进行中	<div>维护删除</div>

共有 9 条数据

1

9 条/页

图 4-8

提示：在维护页面，3 个步骤可点击修改相关设置，如下图所示，纯管理项目没有步骤 1

项目信息维护

2 测试流程维护

版本信息维护

测试负责人 *

刘一手 ×

☒ 提交问题

测试人员 *

刘一手 × xin... ×

☐ 测试交叉

互验人员

☐ 分析问题

分析人

☐ 分配问题

分配人

☒ 修改问题

修改人 *

黄一手 × 刘一招 ×

☐ 开发互检

互检人

☒ 分歧仲裁

仲裁人 *

刘一手 ×

☐ 项目关注

关注人

☒ 测试确认

图 4-9

4.7 数据源

维护数据源基本信息，用于接口测试中通过 SQL 定义接口。

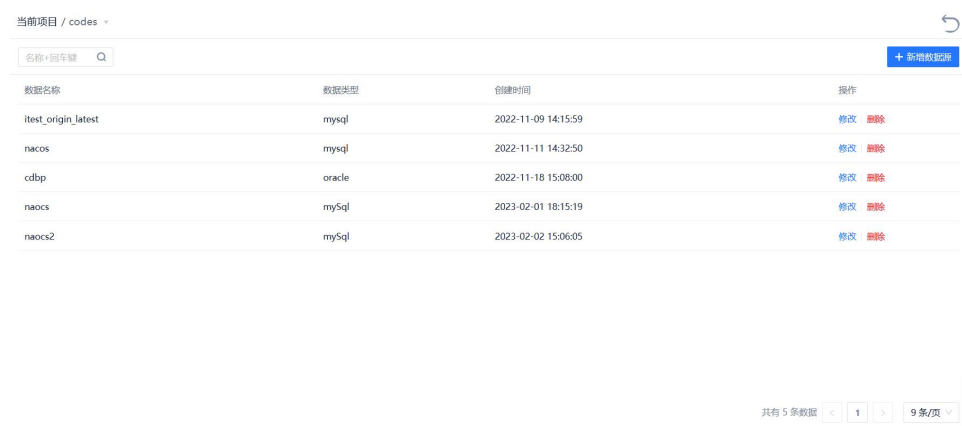


图 4-10

4.8 任务模板

有两个模板，一个是需求任务模板，一个是任务模板。需求任务模板为不同的需求分类建任务模板，当对应需求分类创建需求时，开启任务模板，就会按模板生成相应的任务。维护任务模板基本信息。2.1.0 以上版本，还有一个任务模板，不和需求分类关联，建任务时可以直接选任务模板。

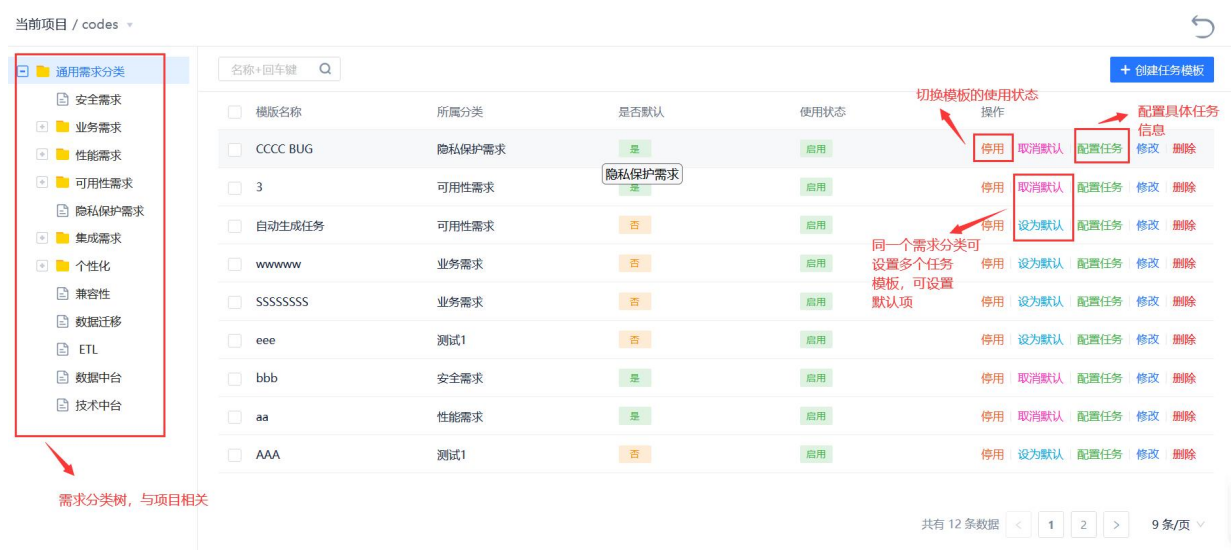


图 4-11

配置任务



图 4-12

4.9 集成配置

4.9.1 安全 key

配置jenkins 和 gitlab 调用 Codes 接口时的 key 值。

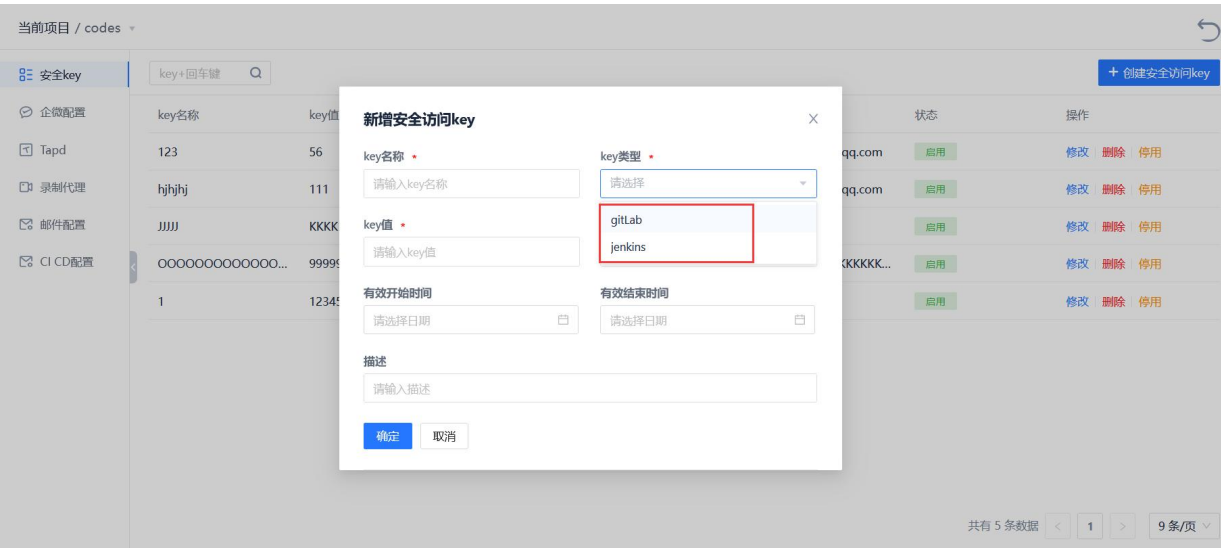


图 4-13

4.9.2 企微配置

配置发送企业微信时的相关参数

当前项目 / codes ▾

安全key

企微配置

Tapd

录制代理

邮件配置

CI CD配置

企业微信配置

企业ID *

应用ID *

应用秘钥 *

确定

图 4-14

4.9.3 Tapd

配置 Tapd 相关参数

当前项目 / codes ▾

安全key

企微配置

Tapd

录制代理

邮件配置

CI CD配置

Tapd配置

API账号 *

API口令 *

公司ID *

确定

图 4-15

4.9.4 录制代理

配置录制代理相关信息，需要安装我们的网关。

当前项目 / codes

安全key

企微配置

Tapd

录制代理

邮件配置

CI CD配置

+ 创建代理服务

<input type="checkbox"/>	服务名称	服务类型	请求方式	代理地址	服务原地址	插入时间	操作
<input type="checkbox"/>	vue	Rest服务	POST	http://127.0.0.1:8280/vue-demo	http://127.0.0.1:8040/vue-demo	2022-05-21 16:59:28	修改 删除
<input type="checkbox"/>	test3	Rest服务	POST	http://127.0.0.1:8280/test3	http://localhost:8080/test3	2022-11-25 15:34:10	修改 删除
<input type="checkbox"/>	test22	Rest服务	POST	http://127.0.0.1:8280/test22	http://localhost:8080/test22	2022-11-25 15:00:29	修改 删除
<input type="checkbox"/>	test5	Rest服务	POST	http://127.0.0.1:8280/test5	http://localhost:8080/test5	2022-11-25 15:53:08	修改 删除
<input type="checkbox"/>	test6	Rest服务	POST	http://127.0.0.1:8280/test6	http://localhost:8080/test6	2022-11-26 15:00:52	修改 删除
<input type="checkbox"/>	SSSSSS	Rest服务	POST	http://127.0.0.1:8280/SSSSSS	122	2023-01-31 19:24:11	修改 删除
<input type="checkbox"/>	test99	Rest服务	POST	http://127.0.0.1:8280/test99	http://localhost:8080/test99	2022-11-26 16:07:26	修改 删除
<input type="checkbox"/>	test100	Rest服务	GET	http(s)://codes.cn	http://localhost:8090	2022-11-26 16:08:42	修改 删除
<input type="checkbox"/>	codes	Rest服务	GET	http(s)://codes.com	http://localhost:8032	2022-11-26 16:09:29	修改 删除

共有 9 条数据

< 1 >

9 条/页

图 4-16

4.9.5 邮件配置

配置邮件发送端相关信息

当前项目 / codes

安全key

企微配置

Tapd

录制代理

邮件配置

CI CD配置

邮件配置

smtp发件服务器 *
smtp@codeswork

port *
400

username *
smtp@codeswork

password *
123456

确定

图 4-17

4.9.6 CI CD 配置

目前仅支持安装 Codes 专业版后，该配置才有效。Codes 的 CICD 插件内置了一些配置，外部的 jenkins 可能没有 sonqr queb 等。配置 Codes 的 CI CD 所关联的 jenkins 相关信息（必须使用 Codes 的 CIDE 的插件）。

图 4-18

5、附录

5.1. Codes 通用操作说明

5.1.1. 我待处理的缺陷

指流转到本人名下的缺陷，即“我的待办”，另外也包含 5 分钟内本人刚新提交的缺陷。点击“缺陷管理”时，缺省只显示流转到我的缺陷，省得被其他缺陷信息“纷扰”。要查看其他缺陷，可以通过查询来查看。

5.1.2. 操作图标

5.1.2.1. 人型图标

指可以查询更多的人员选择项



5.1.2.2. 树型图标

表示点击选择需求



5.1.2.3. 列表切换图标

表示切换列表的显示模式



5.1.3. 公共用例库与产品用例库的区别

公共用例库存放的是所有共性的用例，如登录等之类的，可以从项目中，把认为有共性的用例推荐到公共用例库中，超级管理员审批通过后，正式进入公共用例库。产品用例库，是某个产品的用例库，可以把个在做的项目导入到产品用例库，在导入的时候，会连同测试需求项（也就是模块树）和全部用例一起

导入，在项目中从产品用例库中导入用例时，可以选导入部分或者所有用例，不管导入部分用例还是全部用例，都会先导入所有目录，已有的目录不会重复导入。项目中选从公共用例库中导入用例时，不会导入目录。

5.1.4. 业务场景与迭代的关系

业务场景，用于流程类的业务功能测试，可以复用多个单用功能用例为一个场景，等同于其它同类工具的用例集，只是目的不一样。迭代相当于测试计划。

5.1.5. 升级及备份

Docker 及 Docker-compose 版本，需要停止服务，把 CODES_HOME 下所有目录进行备份，升级时会自动备份数据

5.1.6. gitlab 与 Codes 联动配置

主要用于提交代码和缺陷，需求，任务关联

只需要在 gitLab 配置好 webhook 即可，Codes1.1.0GA_u5 及以上版本才支持该功能。下图中的 url 后的 token 需要在 Codes 的设置菜单下，集成配置中，安全访问 Key 中进行配置。**注意：下图中 URL: getGitInfo 要改为 gitLabCallBackCodes, itest 要改为 codesBackend。**

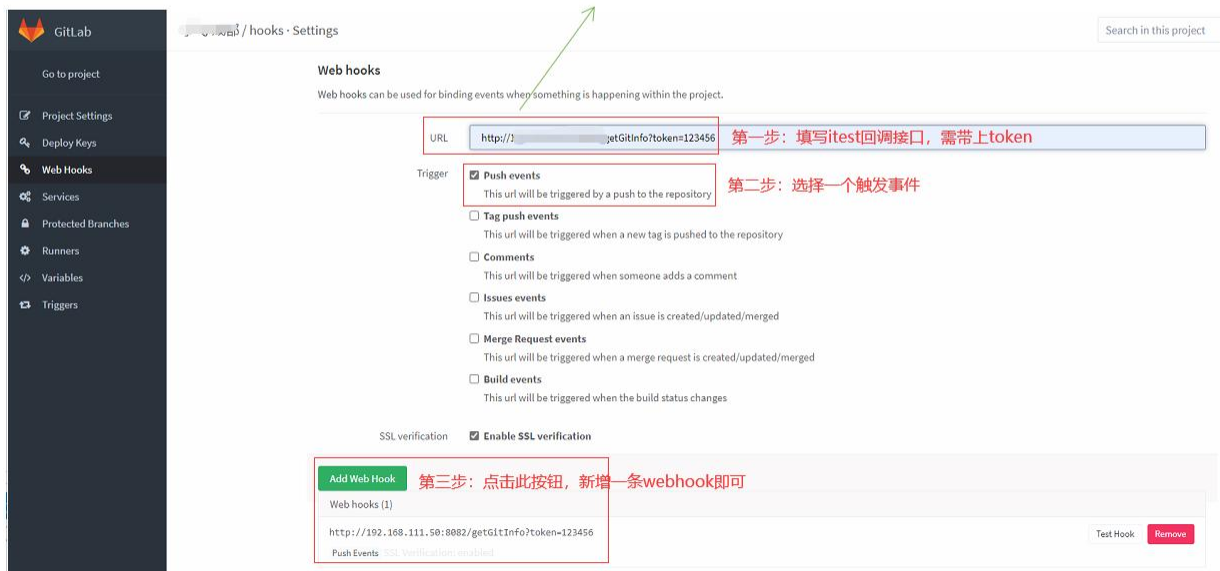
url 为: <http://itest服务器IP:端口/itest/getGitInfo?token=123456>

Docker 示例

<http://192.168.1.100:8010/itest//getGitInfo?token=123456>

windows 示例

<http://192.168.1.100:8091/itest//getGitInfo?token=123456>

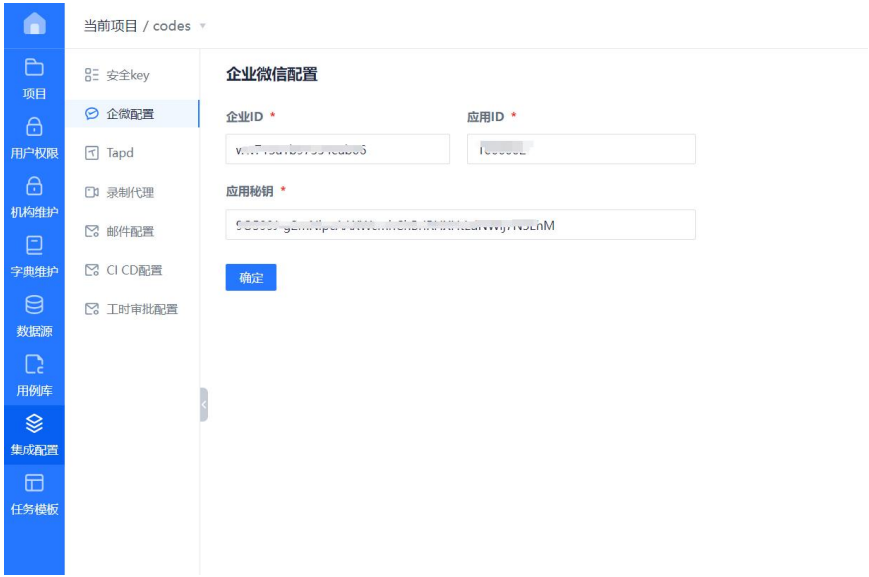


gitLab 中需要同步commit 的项目, 每个项目都需要加同样的web hook

5.1.7. Codes 与第三方 IM 联动配置

5.1.7.1. 与企业微联动

需要每个 Codes 帐号进行关联的企业微信帐号配置, 同时需要 Codes 服务器 IP 所属域名加入到企业微信的白名单中, 否则无法联动信息。同时需要安装 Codes 的服务器配置了域名且在企微后台把这个域名设置到白名单中。



5.1.7.2. 与飞书联动

同钉钉配置类似，机器人配置如下图所示：



然后在项目基本信息中，配置飞书机器人地址

1 项目信息维护

测试流程维护

版本信息维护

项目编号 *	test01	项目状态	进行中
项目名称 *	jmeter 集成	缺陷修复必须关联代码提交	<input type="checkbox"/>
研发部门 *	test01	需求粒度(天)	
开始日期 *	2023-02-13	需求层级	
结束日期 *	2023-08-01	网关上下文	
项目人员 *	陈一招 × 刘一招 × 黄一手 ×		
产品人员 *	黄一手 ×	PM姓名 *	黄一手
飞书地址		钉钉地址	
需求流转发邮件	<input type="checkbox"/>	任务发邮件	<input type="checkbox"/>
关联Tapd项目		执行单位成本	

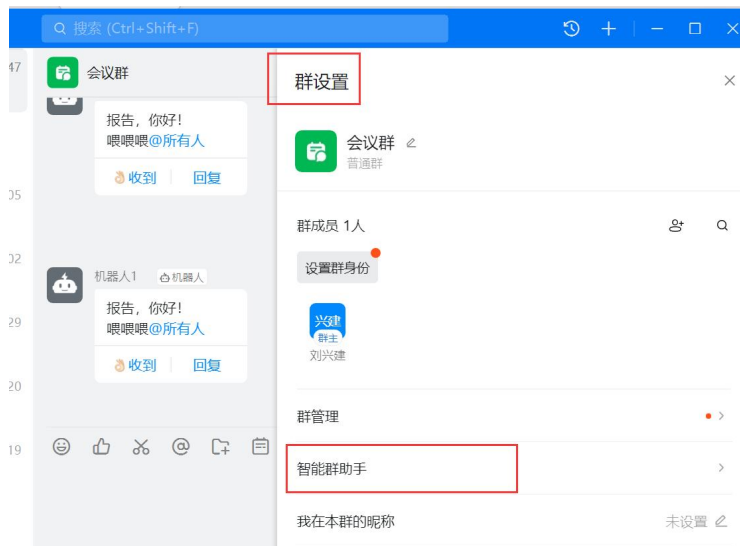
确定

重置

5.1.7.3. 与钉钉联动

5.1.7.3.1. 配置钉钉机器人及其关键词

进入钉钉群，配置钉钉机器人，如下图所示，进入群设置，点击智能群助手。



添加群机器人，选择自定义。



设置机器人名称，进行安全设置（目的在于限制机器人进行消息发送，例如：选择关键词，则只有包含关键词的信息才会进行发送），关键词必须是缺陷或者摘要，如有任务或者用例包也发消息就要再增加关键词：任务、测试包、用例。

5.1.7.3.2. 获得钉钉机器人地址

配置完成后，会得到一个 webhook 地址，如果需要 Codes 向该群推送消息，则复制 webhook 地址到 Codes 项目中的基本信息的钉钉地址中，就可以通过 Codes 向钉钉推送群消息。

5.2. 安装 FAQ

1.

