**GTP**

**详细设计说明书**

(内部资料 请勿外传)

|  |  |  |  |
| --- | --- | --- | --- |
| **编 写：** |  | **日 期：** |  |
| **检 查：** |  | **日 期：** |  |
| **审 核：** |  | **日 期：** |  |
| **批 准：** |  | **日 期：** |  |

**国美在线**

**文档变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **变更（+/-）说明** | **作者** | **版本号** | **日期** | **批准** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[一、 引言 1](#_Toc424919430)

[1、 编写目的和范围 1](#_Toc424919431)

[2、 术语表 1](#_Toc424919432)

[3、 背景 1](#_Toc424919433)

[4、 项目优势 1](#_Toc424919434)

[二、 总体设计 2](#_Toc424919435)

[1、 架构设计 2](#_Toc424919436)

[2、 主服务流程设计 4](#_Toc424919437)

[三、 模块设计 6](#_Toc424919438)

[1、 用例图 6](#_Toc424919439)

[2、 功能模块说明 6](#_Toc424919440)

[2.1 gtp-common模块 6](#_Toc424919441)

[2.2 gtp-domain模块 7](#_Toc424919442)

[2.3 gtp-maven-plugin模块 10](#_Toc424919443)

[2.4 gtp-service模块 13](#_Toc424919444)

[2.5 gtp-web模块 17](#_Toc424919445)

[四、 重要接口设计 20](#_Toc424919446)

[1、 createTask接口 20](#_Toc424919447)

[1.1 接口说明 20](#_Toc424919448)

[1.2 调用方式 20](#_Toc424919449)

[2、 字典表查询接口 20](#_Toc424919450)

[2.1 接口说明 20](#_Toc424919451)

[2.2 调用方式 20](#_Toc424919452)

[3、 Jenkins封装接口 20](#_Toc424919453)

[3.1 接口说明 20](#_Toc424919454)

[3.2 调用方式 20](#_Toc424919455)

[五、 数据库设计 21](#_Toc424919456)

[1、 Mysql 21](#_Toc424919457)

[2、 MongoDB 21](#_Toc424919458)

[六、 系统安全保密设计 21](#_Toc424919459)

[1、 说明 21](#_Toc424919460)

[2、 设计 22](#_Toc424919461)

[七、 系统出错处理 22](#_Toc424919462)

# 引言

## 编写目的和范围

本详细设计说明书编写的目的是说明GTP自动化测试平台的架构设计，包括模块划分、模块功能、执行流程等，为软件编程和系统维护提供基础。本说明书的预期读者为系统设计人员、软件测试人员和项目评审人员。

## 术语表

|  |  |  |
| --- | --- | --- |
| **序号** | **术语或缩略语** | **说明性定义** |
| 1 | GTP | Gome Test Platform |
| 2 |  |  |
|  |  |  |

## 背景

随着敏捷开发思想的日益流行，缩短的开发周期、频繁的部署上线以及复杂的团队组织等问题都给保证软件质量的测试提出了更多挑战。

虽然现有的自动化测试框架已经极大地提高了测试效率，但框架的设计和使用都带有比较鲜明的针对性特点，测试环境配置、测试结果统计、测试用例筛选等一系列现实需求问题仍存在改进空间。

针对以上涉及的类似问题和自动化测试的现实需求，提出了以下解决方案：集成自动化测试框架和Jenkins搭建一套可灵活定制的自动化测试平台，即国美自动化测试平台——GTP。GTP充分利用了现有自动化测试框架的可移植性和Jenkins服务的稳定性、可定制化、分布式等特性。

## 项目优势

基于成熟的自动化测试工具。API自动化测试框架已发布多个稳定版本，定制化的设计特点为API测试提供了良好的支持。GUI、LOAD等自动化测试框架将相继上线，会为UI和性能测试提供支持。这些测试框架都具有高可移植性，GTP为这些自动化测试框架提供了统一的支持。

灵活的定制化测试。测试成员可按需求自助式配置自动化测试工程，目标测试环境，测试集，定时任务等。测试任务将自动执行，并发送邮件通知。

测试报告生成和保存。所有的自动化测试记录都在平台上统一记录，用户可随时查阅，并可按团队或个人查询任意时段的用例数量和通过率报表。

用户体验良好的Web页面访问。为方便测试人员使用，GTP提供了一整套的Web页面，用户要使用GTP执行测试任务需要做的只是填写几项配置。另外，测试报表也使用Web图表的形式直观展现。

Jenkins的成熟的平台。Jenkins提供稳定的分布式服务，执行步骤的自由定制，完善的API接口，邮件提醒等强大的后台支持。

# 总体设计

## 架构设计

GTP通过设计严密的服务流程，为用户灵活定制测试任务提供了体验良好的Web界面，自动将SVN版本控制下的测试用例装配成可执行的测试任务，发布到Jenkins节点机。依赖Jenkins强大的分布式构建策略，提高测试执行效率。同时将测试结果重新进行格式化后落入MongoDB，通过定时聚合策略，用户可通过Web界面即时得到所需测试报表。详细的架构设计如下图所示：

MySQL

GTP WEB

Jenkins

Dispatcher Service

MongoDB

编辑，查看Task，及Task相关配置；展示报表；Task相关数据与Mysql交互，报表相关数据从MongoDB读取

存储汇总后与报表相关的数据

扫描定时/手动触发Task；读取Task，组装为Jenkins Job并触发；监控Job状态；收集并处理Job结果，数据分别落入mysql和mongoDB

Task相关配置；Task及Case 执行详细结果；Schedule、字典等其他记录

根据Service分发的Job参数加载并执行Job；Maven插件统计并返回Job结果；组织维护Agent Cluster

6、Web从MongoDB读报表，TaskResult详情等数据，并展示

5-2、 Service汇总报表相关数据，存入MongoDB

7、从Mysql读取已有Task信息加载到页面，供查看编辑

1、从页面提交数据到Mysql（增删改）

2、扫描Mysql中Task的日程，唤醒Task，加入Task队列；扫描Task队列，创建Jenkins Job

5-1、更新Task状态，记录Error Message

4、 向Service反馈Job执行结果（MVN插件）

3、向Jenkins发送并触发Job，迁出代码执行测试

#### GTP总体架构图

SVN

## 主服务流程设计

GTP主要分为五个独立的服务模块：

GTP Service是GTP的测试任务分发服务，负责根据Mysql数据库中的任务状态和队列状态流转测试任务，最终发送到Jenkins Service；

Jenkins Service负责执行构建测试任务，并更新任务状态，其中还需要使用三个设计好的maven 插件进行构建测试之外的辅助工作；

GTP Web负责用户交互，包括配置测试任务，查看任务状态，展示测试任务报表等工作；

另外还有两个数据库服务，其中Mysql服务主要用于存储基本的数据模型数据，任务状态和待执行任务队列等配置信息；MongoDB负责存储大数据量的测试任务结果的历史信息，并通过定期聚合将数据存入和测试报表相一致的集合中，满足用户即看即用的报表生成需求。

多个服务之间除数据访问之外，没有直接的耦合关系，可独立运行和维护升级，详细的流程如下图所示：

#### GTP 服务流程

Executors Initializer

TaskInfo

TaskList

CreateTask

Executor

AssignTask

Executor

CreateJob

Executor

Waiting

Splitted

Assigned

Sent

Running

Completed

SplitTask

Executor

Arrived

Maven

Plugin

gtp:start

Maven- Plugin

gtp:end

Maven- Plugin

gtp:splitCase

Maven- Plugin

GTP Service

Jenkins Service

MySQL

 Jenkins

Build-Test

Error

TaskResult

MongoDB

Regular

SvnInfo

Machine

Task Configure

Task Report

GTP

WEB

Task/Job

Status

LongSent

Executor

Expire

Executor

Aborted

# 模块设计

## 用例图

gtp-common

gtp-domain

gtp- service

gtp-maven-plugin

gtp- web

GTP使用Maven的父子模块方式构建开发，各模块之间的依赖关系如上图所示。

其中gtp-common模块主要维护整个工程的常量和静态公共方法；gtp-domain模块维护整个工程的数据库访问、数据模型定义、数据相关的逻辑方法；gtp-maven-plugin模块主要编写了三个maven插件用于测试工程构建过程中进行辅助操作；gtp-service模块是整个工程的主服务模块，负责推动测试任务的流程流转；gtp-web模块是GTP与用户交互的Web服务，负责用户对测试任务的配置查看等功能。

## 功能模块说明

### gtp-common模块

Common模块主要用于统一存放GTP工程的常量和静态方法。

常量主要包括字典表的类型关键字、测试任务状态值、配置文件替换标记、超时阈值等。

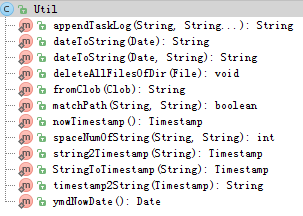
静态方法包括字符串替换处理、日期格式化、文件操作等。

#### 源程序文件说明

Constant.java文件中定义静态常量。

Util.java文件中定义静态方法。

#### 主要函数说明



1. public static void deleteAllFilesOfDir(File path)：删除path文件夹下的所有文件，Java的delete()函数只能删除文件或空文件夹，不支持删除非空文件夹。此函数通过递归删除，达到删除path文件夹内所有文件夹和文件的目的。
2. public static boolean matchPath(String setUrl, String url)：路径匹配函数，支持“\*”通配符进行路径匹配。如“/task/\*”路径，表示“/task”路径下的任何路径都匹配。用于gtp-web模块的拦截器对路径进行过滤。

### gtp-domain模块

gtp-domain模块主要分为以下四个子模块：model、dao、jenkins、schedule

#### Model子模块

##### 功能描述

Model子模块主要定义GTP工程所有的数据模型以及数据库表字段映射关系。

#### Dao子模块

##### 功能描述

Dao子模块主要负责数据库访问操作，包括mysql数据库和MongoDB数据库。

##### 输入数据

查询条件，更新数据。

##### 输出数据

数据库查询结果，更新结果。

##### 业务算法和流程

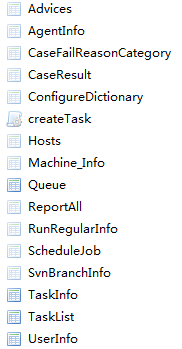
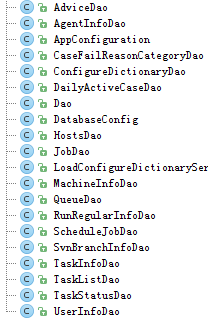
Mysql数据库作为GTP基本数据模型对应的数据库，存储GTP各模块中的模型。绝大多数业务流程的数据交互都由Mysql数据库支持，因此Dao子模块的绝大多数操作都与Mysql数据库交互，包括增删改查等一般的事务操作。

MongoDB用于存放大量的测试报告，Dao子模块的另一个业务流程是存储和读取MongoDB中的测试报告，主要包括增添，读取，聚合等操作。

其中，Mysql库中有一个字典表，用于存放各种字典类信息，该表修改频率较低，但读取频率极高，为减少读取次数，为该表设置了缓存机制，定时更新缓存中的数据。

##### 数据设计

Mysql中的数据表和数据访问类分别如下图所示：

#### Schedule子模块

##### 功能描述

Schedule子模块用于日程流转中的数据访问操作，如更新任务状态，将任务插入待执行队列等操作。

##### 输入数据

任务ID、原状态、当前状态、时间戳

##### 输出数据

更新结果

##### 业务算法和流程

主要包括更新，查询，插入操作

##### 数据设计

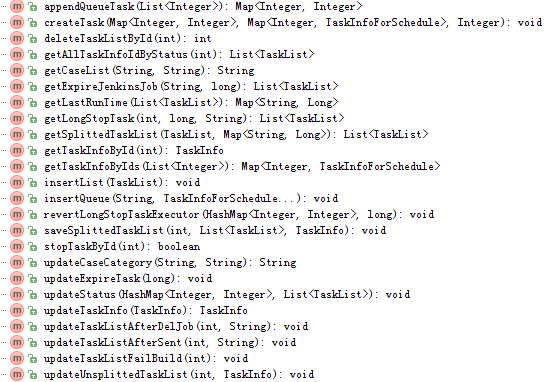
HashMap<cronExp,List<taskId>>

该哈希映射作为定时任务的运行依据，CronExp表示定时任务的CronExpression，List<taskId>则表示定时时间是CronExp的所有定时任务的ID。通过这样一种数据结构可以按Cron表达式一次性将按定时需要执行的任务加入到待执行队列中。

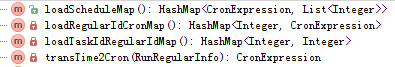
这一数据结构需要通过多表查询以及定时转换等一系列操作获取。

##### 函数说明

Schedule子模块的函数如下：



ScheduleJob中主要包括数据库操作的增删改查几种类型的函数。



ScheduleBo中主要包括上述哈希映射关系的获取操作函数。

#### Jenkins子模块

##### 功能描述

Jenkins子模块用于和Jenkins的Master机通信，通过对Jenkins服务的API进行深度封装实现对Jenkins某些功能的调用。

##### 输入数据

Hosts文件内容，测试任务信息，Config.xml配置文件模板

##### 输出数据

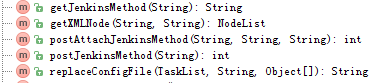
向Jenkins发送POST或GET请求

##### 业务算法和流程

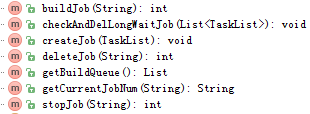
GTP服务调用Jenkins子模块，通过将测试任务相关的参数，类型选择相应的config.xml文件进行内容替换，最终作为调用Jenkins API 接口的POST请求的附件发送到Jenkins，驱动Jenkins服务。

##### 函数说明

Jenkins子模块的核心函数如下：



包括向Jenkins发送POST、GET请求的封装函数，获取Jenkins API XML节点的函数和替换config.xml模板文件的函数。



还包括操作Jenkins构建Job和构建队列的API封装函数。

### gtp-maven-plugin模块

gtp-maven-plugin模块包括gtp:start、gtp-splitCase、gtp-end三个maven插件，用于自动化测试构建过程中的额外操作。

Running

Completed

gtp:start

gtp:end

gtp:splitCase

 Jenkins

Build-Test

Error

#### gtp:start插件

##### 功能描述

在测试任务构建之前执行，用于更新测试任务状态为Running，表明该测试任务已经被成功的发送到节点机且开始执行。

##### 输入数据

测试任务ID、测试任务原状态。

##### 输出数据

更新测试任务状态为Running或更新失败。

##### 业务算法和流程

当测试任务发送到节点机后，测试构建执行的第一步为执行gtp:start插件，更新测试任务状态。

##### 函数说明

重载Maven插件的execute()函数，先判断测试任务的原状态是否合法，然后更新测试任务的状态为Running或Aborted。

##### 限制条件

测试任务的状态必须由合法的状态更新到下一状态，当测试任务的状态更新过程非法时应立即将其状态更新为中止，禁止跨阶段更新状态。

#### gtp:splitCase插件

##### 功能描述

在测试任务构建之前执行，用于根据测试任务拆分后的结果删除某些不需要执行的测试用例，防止多个子任务重复执行相同测试用例。

##### 输入数据

测试任务ID、测试用例列表。

##### 输出数据

删除重复测试用例后的测试用例列表。

##### 业务算法和流程

当测试任务开始执行后，需要先根据拆分结果删除该子任务分配到的测试用例之外的重复用例，具体做法为根据输入列表删除testng.xml中的某些用例节点。

##### 函数说明

重载Maven插件的execute()函数，先判断测试任务的原状态是否合法，然后删除重复测试用例。

##### 限制条件

只有当前测试任务为拆分后的子任务才需要执行，未拆分过的测试任务不需要执行。

#### gtp:end插件

##### 功能描述

在测试任务构建之后执行，用于更新测试任务状态为某结束状态，表明该测试任务已经被成功的执行过。具体的结束状态需要根据任务最终的实际状态决定。

同时还需要将测试报告存入MongoDB。

##### 输入数据

测试任务ID、测试任务原状态、测试任务实际状态、测试报告json文件。

##### 输出数据

更新测试任务状态为结束状态，测试报告存入MongoDB。

##### 业务算法和流程

gtp:end插件会根据构建结果获取到测试任务实际的最终状态，然后根据任务ID更新对应任务的状态。测试任务成功执行后还会生成测试结果的json文件，gtp:end插件需要将该文件内容存入MongoDB，用于测试结果的统计。

##### 函数说明

重载Maven插件的execute()函数，先获取测试任务最终的实际状态，然后更新该测试任务的状态，以及更新MongoDB中的测试结果。

##### 限制条件

只有成功的测试任务才会生成测试结果的json文件。

### gtp-service模块

gtp-service模块作为GTP的主服务模块，负责GTP测试任务的流程控制，该模块根据数据库的状态能够及时作出相应处理，控制测试任务进行正常流转。该模块主要包括七个任务执行器和两个辅助类。

CreateTask

AssignTask

CreateJob

Waiting

Splitted

Assigned

Sent

SplitTask

Arrived

RevertLongSent

AbortExpire

Aborted

DeleteJenkins

#### CreateTask

##### 功能描述

将需要执行的测试任务添加到待执行队列中，测试任务可以是定时任务也可以是手动触发的任务。

##### 输入数据

##### Map<CronExpression, List<Integer>>,即Schedule子模块中定义的CronExp和TaskID的哈希映射关系

##### 输出数据

待执行测试任务队列

##### 业务算法和流程

调用domain模块的loadSchedule()函数，获取哈希映射关系，与当前时间对比，决定该列表中的TaskID是否应该插入到待执行队列。插入队列后状态应更新为Waiting。

#### SplitTask

##### 功能描述

将插入到待执行队列中，状态为Waiting的TaskList条目取出，根据拆分算法将其拆分为多条测试任务，替换原始任务，并将状态更新为Splitted。

##### 输入数据

状态为Waiting的TaskList列表

##### 输出数据

拆分后的子任务或状态为Splitted的不需要拆分的任务列表

##### 业务算法和流程

调用domain模块的getSplittedTaskList()函数，进行测试用例拆分，将一条测试任务拆分为多条，拆分依据为测试用例的历史执行时间，需要访问MongoDB进行获取。

#### AssignTask

##### 功能描述

为拆分完成后的测试任务或不需要拆分的测试任务分配执行机器。

##### 输入数据

状态为Splitted的TaskList列表，当前可用执行机信息。

##### 输出数据

待执行任务队列和分配到的执行机之间的对应关系。

##### 业务算法和流程

不指定执行机的情况下调用domain模块的getIdleMachineLabel()函数，根据测试任务所需环境和当前空闲机器状态为测试任务分配合适的执行机。指定机器的情况下则查看所指定机器是否空闲，决定立即分配或推迟分配。

##### 补充说明

AssignTask阶段实际执行过程穿插到接下来的CreateJenkins阶段，只是逻辑上为一个独立的执行过程。

#### CreateJenkins

##### 功能描述

将拆分完成的子任务或不需要拆分的任务发送到Jenkins节点机，生成Jenkins 临时Job，准备构建测试。

##### 输入数据

拆分后的子任务信息及相关信息、Jenkins config.xml模板文件、

##### 输出数据

向Jenkins发送带有完整信息的Config.xml头文件的POST请求，创建并启动Jenkins临时Job。

##### 业务算法和流程

定时获取待执行队列中状态为Splitted的TaskList条目，逐条根据TaskList的信息反向获取Hosts内容，任务类型、SVN URL等相关信息。根据类型选择相应的config.xml文件模板，并将其中的字段替换为实际内容，通过封装的Jenkins API函数向Jenkins发送带有config.xml头文件的POST请求，创建Jenkins临时Job并激活。

#### RevertLongStopTask

##### 功能描述

将发送到Jenkins一段时间后仍未执行的测试任务重新回退到Splitted状态，并重新分配。

##### 输入数据

发送到执行机的子任务信息及相关信息，即状态为Sent的子任务信息

##### 输出数据

SentToAgentTime超过既定阈值的任务列表。

##### 业务算法和流程

通过定时计算SentToAgentTime找出发送到节点机超过既定阈值时间仍未执行的任务列表，调用Jenkins API将其从构建队列删除并将其状态回退为Splitted，等待下一次重新分配执行机。

#### AbortExpireTask

##### 功能描述

将长时间未结束的测试任务强制设置为Aborted状态

##### 输入数据

所有处于中间状态的测试任务列表

##### 输出数据

StartTime超过既定阈值的测试任务列表

##### 业务算法和流程

从插入待执行队列状态更新为Waiting开始，到某一结束状态之前的所有测试任务都处在中间状态。长时间处于中间状态而未结束的任务认为是异常现象，应及时强制将其置为Aborted状态。

#### DeleteJenkins

##### 功能描述

清理Jenkins上过期的临时Job

##### 输入数据

Jenkins Job名称、执行时间

##### 输出数据

Jenkins API 删除Job的响应结果

##### 业务算法和流程

发送到Jenkins的Job都是临时Job，只运行一次。下次相同的测试任务会重新创建Jenkins Job。因此，一段时间后执行机会产生大量的历史Job，影响机器性能，需要定期清理。通过封装Jenkins API的接口可以批量快速删除Jenkins Job及其构建历史。

#### UpdateTaskReport

##### 功能描述

聚合MongoDB中的测试报告数据，供GTP Report快速生成。

##### 输入数据

一定日期间的所有测试报告数据

##### 输出数据

聚合的带有日期、群组、所有人索引的汇总测试报告数据

##### 函数说明

1. public void updateReportOwner(int minDate, int maxDate, int taskType)：

以所有人为索引，按测试类型更新minDate到maxDate之间的测试报告数据，并存入MongoDB的汇总表。

1. public void updateReportGroup(int minDate, int maxDate, int taskType, Map<Integer, Integer> taskIdAndGroupId)：

以群组为索引，按测试类型更新minDate到maxDate之间的测试报告数据，并存入MongoDB的汇总表。

##### 业务算法和流程

GTP Report需要按不同维度生成测试报告，当测试数据过多时，产生一份详细的测试报告比较费时。通过定期将测试报告计算汇总，GTP Report每次可立即取到汇总后的数据，可以实现报告的立即生成。

### gtp-web模块

gtp-web模块作为用户操作GTP的Web入口，主要为用户提供友好的测试任务配置查看页面，测试任务相关子模块配置查看页面，GTP报表查看页面等前端功能。同时，还可以通过手动触发、中止测试任务，实时查看测试任务运行状态和用例级的详细测试报告。主要分为Task、SVN-Branch、Run Rule、Machine、Hosts、GTP Report六个功能模块，以及权限设置管理等其他非功能模块。

#### Task

##### 功能描述

GTP中每一条测试任务都被配置成一条Task，其信息直接作为测试任务的执行依据。每条Task可指定SVN地址，测试环境，定时任务等相关信息；同时可以查看所有已执行过的测试任务详细结果，最小颗粒度为一条测试用例的执行结果。

Task模块是以其他模块的详细配置为基础的，这是因为一条Task包含的信息过多，如果集中配置信息比较繁杂，将每个子模块已配置好的条目作为Task执行的选项便于细分功能模块和降低模块间的耦合度。

##### 业务算法和流程

GTP主服务只与Task直接交互，包括测试任务的相关信息最终都会整合为一条Task。因此，Task需要和其他子模块相关联，通过数据库级联访问等途径获得一条测试任务的所有详细信息。同时Task的状态会作为gtp-service模块流程流转的依据。

#### SVN-Branch

##### 功能描述

每个测试团队都需要有统一的存放自动化测试用例的SVN分支，本团队所有测试人员都需要从该地址迁出、更新测试用例。

需要将该SVN地址配置到GTP中，每一条测试任务都需对应一个SVN地址，以便执行自动化测试过程中自动迁出测试用例。

一条测试任务归属于哪个团队，最终也是通过该测试任务所使用的SVN地址的分支所归属的团队决定的。

##### 业务算法和流程

GTP执行测试任务工程中多需要的任何测试代码，测试用例都以SVN上的最新版本为准，每次都会迁出最新的版本执行。因此用户的所有相关代码和用例都需要及时提交SVN才能够正确执行。

#### Run Rule

##### 功能描述

每条测试任务支持运行规则的灵活定制，用户可以按需要配置一条运行规则，可以配置为每天、每周或仅一次等执行规则。

GTP要根据测试任务的运行规则，定时自动执行测试任务。

##### 业务算法和流程

用户自定义的Run Rule最终会由gtp-domain模块中的Schedule子模块转换成Cron表达式，统一维护日程。为方便用户设置，该模块并没用要求用户直接输入Cron表达式，而是采用更直观的时分秒时间供用户选择。

#### Machine

##### 功能描述

某些测试任务需要特殊的运行环境，所有当前可用的执行机配置信息都要添加到Machine模块的机器列表中，供用户选择。当测试任务不指定机器时，GTP将根据当前可用机器状态和任务执行历史为测试任务自动分配机器。

该模块需要管理员进行维护，用户只需按要求从中选取。

##### 业务算法和流程

用户可以为测试任务指定Machine列表中的节点机作为执行环境，也可以由系统自动分配。实际上这里的机器列表为Jenkins的节点机，只不过添加了节点机的配置信息，供用户按需要筛选。Jenkins API提供了指定节点、指定集群、不指定节点几种方式。

#### Hosts

##### 功能描述

不同测试环境的测试任务需要在测试执行前修改执行机的Hosts文件，用户可以按环境统一维护一条Hosts配置，并在测试任务中指定测试环境。测试执行前，GTP应先自动更新执行机的Hosts文件，然后执行测试任务。

##### 业务算法和流程

同构的测试环境会分配不同的网段，相同的服务在不同的测试环境会有不同的访问hosts ip，因此需要在执行测试之前修改执行机的Hosts。另一方面，某条测试任务属于哪个测试环境是由该任务所采用的hosts决定的。

#### GTP Report

##### 功能描述

GTP上运行过的测试任务需要将测试结果进行统计，并直观展现给用户。可以按测试类型、个人、团队查询总Case数和测试通过率。

##### 业务算法和流程

gtp-service模块的UpdateTaskReport子模块定期聚合的测试结果数据，最终会供GTP Report模块直接调用，向用户展示多维度直观的曲线图报表。用户可自主按条件，按时间段，按类别查询所需报表。该模块支持最小的颗粒度为某人某天执行的用例数或通过率。

#### 权限管理

##### 功能描述

GTP某些模块只允许管理员操作，这主要考虑到测试任务的稳定运行，不允许普通用户更改Machine，Report生成等。

##### 业务算法和流程

权限主要通过添加三层FilterRegistrationBean，将不同角色用户行为进行过滤和处理。

其中SimpleFilter会允许一般用户操作，又将一般用户的登录状态和非登录状态进行分别处理。非登录状态下允许用户进行所有的非编辑操作，登录状态下除少数模块外可以进行任何操作。

AdminFilter模块会处理SimpleFilter拦截的操作，当用户为管理员身份登录时会允许用户进行所有模块的所有操作。

# 重要接口设计

## createTask接口

### 接口说明

创建一条测试任务需要通过获得任务来源和配置信息，加入待执行队列。

### 调用方式

public void createTask(Map<Integer, Integer> objectQueue,

Map<Integer, TaskInfoForSchedule> taskInfoList, Integer taskId)

通过对前两个参数进行判空操作，可以判断测试任务是来自手动触发还是定时任务，第三个参数直接给出任务的ID。最终测试任务将被加入手动触发队列或定时任务队列。

## 字典表查询接口

### 接口说明

字典表存储了GTP项目的大量常量键值对对应关系，需要频繁的查询此表。通过缓存策略将三级字典表存储，设计了正反两个接口进行键、值的查询。

### 调用方式

public String getNameByValue(String type, Integer value):

根据类型和值可以查询到常量名称；

public Integer getValueByName(String type, String name)

根据类型和名称可以查询到常量值；

## Jenkins封装接口

### 接口说明

Jenkins API提供了基本的调用接口，根据GTP工程的特殊性，进行了二次封装，可以直接操作GTP与Jenkins交互。

### 调用方式

public void createJob(TaskList taskList):

根据任务信息创建Jenkins Job，通过头文件中附加config.xml文件的方式发送POST请求，创建Jenkins 临时 Job。

public int buildJob (String jobName):

根据Job名称启动构建Job，根据返回的HTTP CODE判断结果。

public int stopJob (String jobName):

根据Job名称停止构建Job，根据返回的HTTP CODE判断结果。

public int deleteJob (String jobName):

根据Job名称删除构建Job，根据返回的HTTP CODE判断结果。

public void checkAndDelLongWaitJob (List<TaskList> longWaitTaskList):

查询任务列表中长时间等待的Job，并强制删除，用于gtp-service模块中AbortExpireTask做后续清理工作时调用。

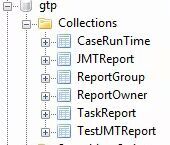
# 数据库设计

## Mysql

如gtp-domain模块的Dao子模块所述，Mysql数据库作为GTP基本数据模型对应的数据库，存储GTP各模块中的模型。绝大多数业务流程的数据交互都由Mysql数据库支持，Dao子模块的所有除报表之外的数据访问都来自Mysql数据库。

## MongoDB

考虑到测试结果内容详细且数量增长速度极快，使用了适合存储json格式测试报告的MongoDB。根据GTP报表的查询维度和查询条件最终会将MongoDB中的数据汇总到以下集合中，通过为每个集合建立索引加快报表的生成速度。



# 系统安全保密设计

## 说明

由于GTP系统需要按个人或团队进行用例数量和通过率的统计，且测试任务按团队区分，需要考虑操作的安全性问题。因此需要用户登录才能进行编辑操作，且所有操作都会依照登录身份信息进行记录。

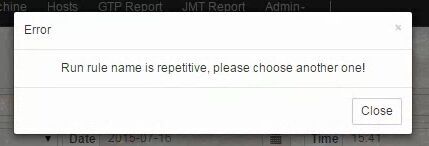
## 设计

主要设计方案为Spring Filter过滤器、SHA-1密码加密、Session判定等技术组合完成。

# 系统出错处理

1. 使用提示性红色字符进行提示，这主要在输入或修改的情况下使用：

2．使用统一的错误界面提示，该界面使用前端Bootstrap框架的模态框。错误界面样式如下图所示：



错误提示界面