



迭代 8 测试计划

面向智慧工厂的准实时监管系统



项目组组号：NO. 1

项目负责人：周汉辰

联系电话：18621966896

电子邮箱：1341634255@qq.com

2016 年 12 月

文档信息

标题	面向智慧工厂的准实时监管系统迭代 8 测试计划
作者	朱一鸣
创建日期	2017/01/01
上次更新日期	2017/01/10
版本	V2. 1
组号	NO. 1

修改历史

日期	版本	说明	作者
2017/01/01	V1. 0	初稿	朱一鸣
2017/01/03	V2. 0	格式修改	曹雨婷、张鹤腾
2017/01/10	V2. 1	内容修改	周汉辰、曹雨婷

目录

一.	简介.....	3
1.	目的.....	3
2.	范围.....	3
3.	目标读者.....	3
4.	文档术语和首字母缩写	3
5.	参考资料.....	3
二.	目标测试项	3
三.	测试计划.....	4
1.	测试类型.....	4
2.	测试环境及配置	4
3.	测试过程.....	4
	3.1 单元测试.....	4
	3.2 集成测试.....	5
	3.3 系统测试.....	5
四.	测试终止条件	6

一. 简介

1. 目的

面向智慧工厂的准实时监管系统迭代 8 迭代测试计划的目的是：

- 提供中心工件，用于管理测试工作的计划和控制。它定义了将用于测试软件和评估测试结果的一般方法，并且是管理者将用来管理和指导详细测试工作的最高级别规划。
- 当在测试工作中需要对管理测试工作的各方面进行全面的关注，并且那些项目干系人适合来批准计划时，为项目干系人提供可见性。

该迭代测试计划也支持以下特定目标：

为总结项目测试阶段的测试情况以及分析测试结果，描述系统是否符合用户需求，是否已达到用户预期的功能和非功能目标制定详细计划。

2. 范围

整个面向智慧工厂的准实时监管系统的所有功能的测试。

3. 目标读者

项目干系人（包括项目的甲方、测试人员、开发人员、项目管理者、其他管理人员等）

4. 文档术语和首字母缩写

无

5. 参考资料

《面向智慧工厂的准实时监管系统需求规约》

《面向智慧工厂的准实时监管系统迭代 8 迭代计划》

二. 目标测试项

面向智慧工厂的准实时监管系统的所有模块，包括：可视化模块、数据推送前置模块、数据组装模块、数据获取模块、配置管理模块的功能和非功能测试。

三. 测试计划

1. 测试类型

ID	测试类型	测试内容	测试目的	工具和方法
1	单元测试	模块关键方法	确保方法的正确性	黑盒测试、边界值测试、手工测试
2	集成测试	各模块集成为整个系统	逐步让各模块合成为整个系统，从而验证整个系统的功能、性能等需求	自顶向下的增量式集成
3	系统测试	测试整个系统的功能执行	验证功能是否符合需求	黑盒测试、手工测试

2. 测试环境及配置

ID	资源名称/类型	配置
1	测试 PC	硬盘 512G，内存 8G，i7
2	测试数据库	MySQL、MongoDB、SQL Server
3	编程软件	Eclipse、WebStorm
4	测试工具	JUnit

3. 测试过程

3.1 单元测试

ID	类名	被测试方法	PASS/FAIL	备注
1	TestMaven	public Object find(String SQL)		源数据 SQL 执行
2	TestMaven	public Object find(String SQL)		源数据 SQL 执行
3	TestMaven	public Object findList(String SQL)		源数据 SQL 执行
4	TestMaven	public Object findList(String SQL)		源数据 SQL 执行
5	TestMaven	public Object findList(String SQL)		压力测试
6	TestMongo	public void get()		HTTPget 请求
7	TestMongo	public void threadTest()		线程测试
8	Test	public void get()		HTTPget 请求
9	Test	public JSON memcachedGet(String name)		Memcached 数据获取请求
10	Test	public void pressureGet()		HTTPget 压力测试
11	TestMongoDao	public ArrayList<SourceData> getAllSource()		获取所有源数据
12	TestMongoDao	public ArrayList<GoalData> getAllGoal()		获取所有目标数据

13	TestMongoDao	public void addSourceDocument (SourceData data)		添加源数据
14	TestMongoDao	public void addGoalDocument (GoalData data)		添加目标数据
15	TestMongoDao	public void deleteSourceDocument (String id)		删除源数据
16	TestMongoDao	public void deleteGoalDocument (String id)		删除目标数据
17	TestMongoDao	public void updateSourceDocument (SourceData data)		更新源数据
18	TestMongoDao	public void updateGoalDocument (GoalData data)		更新目标数据

3.2 集成测试

采用自顶向下的增量式集成测试方法，分别测试各个模块，采用 Stub 代替其他模块，然后逐步使用真实的模块代码替换。

3.3 系统测试

ID	测试描述	输入数据	预期输出	实际输出	PASS/ FAIL	备注
1	对应需求规约，查看各个功能的显示	无	所有数据均能正确显示			
2	对源数据进行配置	{"name": "s_test_1", "type": "single", "tupleNum": "5", "frequency": "1m", "source": "data SourceB", "SQL": "SELECT * FROM congestion WHERE congestion_id=1"}	系统提示“添加成功”			
3	对目标数据进行配置，使用规则 1	{"name": "t_resource_database", "type": "list", "tupleNum": 2, "frequency": 60, "rule": {}, "dataSourceList": [{"name": "s_resource_database", "frequency": 60}]}	系统提示“添加成功”，并且可以从后台成功获取该数据			
4	对目标数据进行配置，使用规则 2	{"name": "t_schedule_list", "type": "list", "tupleNum": 5, "frequency": 10, "rule": {"ruleName": "sort", "key": "num", "order": "desc"}, "dataSourceList": [{"name": "s_schedule_1", "frequency": 60}, {"name": "s	系统提示“添加成功”，并且可以从后台成功获取该数据			

		_schedule_2", "frequency":60}, {"name": "s_schedule_3", "frequency":60}, {"name": "s_schedule_4", "frequency":60}, {"name": "s_schedule_5", "frequency":60}]}				
5	对目标数据进行配置，使用规则 3	{"name": "t_test_3", "type": "single", "tupleNum":2, "frequency":10, "rule": {"ruleName": "exp", "key": "flow", "expression": "s_test_1 + s_test_2"}, "dataSourceList": [{"name": "s_test_1", "frequency":10}, {"name": "s_test_2", "frequency":10}]}	系统提示“添加成功”，并且可以从后台成功获取该数据			

四. 测试终止条件

发现的 bug 全部已修复，100%测试覆盖，且缺陷数收敛。