



迭代 8 测试报告

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

2017 年 01 月

文档信息

标题	面向智慧工厂的准实时监管系统迭代 8 测试报告
作者	朱一鸣
创建日期	2017/01/08
上次更新日期	2017/01/10
版本	V2.0
组号	NO. 1

修改历史

日期	版本	说明	作者
2017/01/08	V1.0	初稿	朱一鸣
2017/01/10	V2.0	文档内容填写	周汉辰、曹雨婷

目录

一.	引言.....	3
二.	测试参考文档.....	3
三.	测试情况.....	3
1.	测试执行情况.....	3
1.1	单元测试.....	3
1.2	集成测试.....	4
1.3	系统测试.....	4
2.	缺陷的统计.....	5
四.	测试结论和建议.....	5

一. 引言

对面向智慧工厂的准实时监管系统的测试情况进行总结和回归测试,描述系统是否符合用户需求,是否达到用户预期的功能目标,并对质量进行分析。

本文档提供给用户、测试人员、开发人员、项目管理者、其他管理员和高层阅读。

二. 测试参考文档

《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 测试计划》。

三. 测试情况

1. 测试执行情况

1.1 单元测试

ID	类名	被测试方法	PASS/FAIL	备注
1	TestMaven	public Object find (Sting SQL)	PASS	源数据 SQL 执行
2	TestMaven	public Object find (Sting SQL)	FAIL	源数据 SQL 执行
3	TestMaven	public Object findList (Sting SQL)	PASS	源数据 SQL 执行
4	TestMaven	public Object findList (Sting SQL)	PASS	源数据 SQL 执行
5	TestMaven	public Object findList (Sting SQL)	PASS	压力测试
6	TestMongo	public void get ()	PASS	HTTPget 请求
7	TestMongo	public void threadTest ()	PASS	线程测试
8	Test	public void get ()	PASS	HTTPget 请求
9	Test	public JSON memcachedGet (String name)	PASS	Memcached 数据获取请求
10	Test	public void pressureGet ()	PASS	HTTPget 压力测试
11	TestMongoDao	public ArrayList<SourceData> getAllSource ()	PASS	获取所有源数据
12	TestMongoDao	public ArrayList<GoalData> getAllGoal ()	PASS	获取所有目标数据
13	TestMongoDao	public void addSourceDocument (SourceData data)	PASS	添加源数据
14	TestMongoDao	public void addGoalDocument (GoalData data)	PASS	添加目标数据
15	TestMongoDao	public void deleteSourceDocument (String id)	PASS	删除源数据
16	TestMongoDao	public void deleteGoalDocument	PASS	删除目标数据

		(String id)		
17	TestMongoDao	public void updateSourceDocument (SourceData data)	PASS	更新源数据
18	TestMongoDao	public void updateGoalDocument (GoalData data)	FAIL	更新目标数据

1.2 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。

1.3 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	所有数据均能正确显示	PASS	
2	对源数据进行配置	{"name": "s_test_1", "type": "single", "tupleNum": "5", "frequency": "1m", "source": "dataSourceB", "SQL": "SELECT * FROM congestion WHERE congestion_id=1"}	系统提示“添加成功”	系统提示“添加成功”	PASS	
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	系统提示“添加成功”，并且可以从后台成功获取该数据	PASS	
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_schedule_2", "frequency": 60}, {"name": "s_schedule_3", "frequency": 60}]}	系统提示“添加成功”，并且可以从后台成功获取该数据	系统提示“添加成功”，并且可以从后台成功获取该数据	PASS	

		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	系统提示“添加成功”，并且可以从后台成功获取该数据	PASS	

2. 缺陷的统计

测试类型	缺陷个数	是否解决
单元测试	2	是
集成测试	0	/
系统测试	0	/

四. 测试结论和建议

质量属性	已完成	建议
功能性	所有功能性需求均完成	
易用性	系统操作界面简洁易用	界面还可以进一步美化
可靠性	基本稳定	
安全性	基本安全	配置管理平台建议单个用户的密码安全保存在配置文件中，建议加密保存