

测试文档

时间	修改者	内容	版本
2021-3-16	冯泊涓	单元测试用例1-2	v1.0
2021-3-21	周瑾瑜	集成测试用例1-11	v1.1
2021-3-21	冯泊涓	单元测试用例3-8	v1.2

单元测试用例

id	用例	覆盖流程	输入	预期输出	测试结果
1	neo4jConnect	neo4jUtil: 连接数据库	无	连接成功	成功
2	insert_delete_Entity	neo4jData: 添加、删除节点	name: "test" age: 1 job: "test"	添加、删除成功	成功
3	insert_delete_Relation	neo4jData: 添加、删除关系	添加节点node1, node2, 关系link 获得node1, node2的 id, 删除关系	添加、删除成功	成功
4	search_Entity	neo4jData: 根据id搜索节点	添加节点node并获取 id, 根据id查找结点, 比较节点的 name,job,age	查找节点与添加的 节点属性一致	成功
5	insert_reverse_delete_Relation	neo4jData: 添加、反转、修改、 删除关系	添加节点node1到 node2的关系link 关系反转 关系改名为new_link 查找节点node2到 node1的关系, 验证关 系名是否为new_link	查找到的 关系名为 new_link	成功
6	get_Child_Entity	neo4jData: 获取节点相关所有关系	添加节点 node1,node2,node3, 添加关系node1- node2, node1- node3, 查找node1的 所有关系	查找到 node1的 关系列表 长度为2	成功
7	get_All	neo4jData: 获取所有节点和关系	添加5个节点、6条关系	查找到节点列表长度为5, 关系列表长度为6	成功
8	update_Entity_Name	neo4jData: 更新实体名字	添加实体node, 改名为 newNode	获取节点名为 newNode	成功

集成测试用例

id	用例	覆盖流程	输入	预期输出	测试结果
1	delETest	删除实体	实体ID	实体被删除	成功
2	importJsonTest	导入Json文件解析并持久化到数据库	Json字符串	导入成功, neo4j数据库持久化成功	成功
3	getAllTest	从数据库中调取所有的实体	无	将数据库中实体全部返回	成功
4	getChildETest	得到一个实体的所有子实体	实体ID	返回一个实体的所有相关子实体	成功
5	addETest	增加一个新的实体并持久化到数据库	实体属性	返回实体的ID	成功
6	addRTest	增加一个新的关系并持久化到数据库	关系所属实体及关系内容	返回关系的ID	成功
7	updENAME	修改一个已有的实体名称并持久化到数据库	实体的ID和新名字	实体的名称被修改成新名称	成功
8	updrNameTest	修改一个已有的关系名称并持久化到数据库	关系的ID和新名字	关系的名称被修改成新名称	成功
9	rev_r_dir_test	倒转一个已有的关系两端实体并持久化到数据库	关系的名字和两端实体的ID	关系两端实体被倒转	成功
10	del_e_test	删除一个已有的实体并持久化到数据库	实体的ID	数据库中该实体被删除	成功

id	用例	覆盖流程	输入	预期输出	测试结果
11	del_r_test	删除一个已有的关系并持久化到数据库	关系的两端节点ID和关系名称	数据库中该关系被删除	成功