**activity工作流**

Activity工作流的学习，想通过Activity工作流的形式去整合ELK、MQ和SpringCloud。

**一、什么是工作流**

以请假为例，现在大多数公司的请假流程是这样的： 员工打电话（或网聊）向上级提出请假申请—上级口头同意— 上级将请假记录下来—月底将请假记录上交公司—公司将请假录入电脑

采用工作流技术的公司的请假流程是这样的： 员工使用账户登录系统—点击请假—上级登录系统点击允许

就这样，一个请假流程就结束了 有人会问，那上级不用向公司提交请假记录？公司不用将记录录入电脑？

答案是，用的。但是这一切的工作都会在上级点击允许后自动运行！ 这就是工作流技术。 Georgakopoulos给出的工作流定义是： 工作流是将一组任务组织起来以完成某个经营过程： 定义了任务的触发顺序和触发条件，每个任务可以由一个或多个软件系统完成， 也可以由一个或一组人完成，还可以由一个或多个人与软件系统协作完

**二、工作流技术的优点**

从上面的例子，很容易看出 工作流系统，实现了工作流程的自动化，提高了企业运营效率、 改善企业资源利用、提高企业运作的灵活性和适应性、提高量化考核业务处理的效率、减少浪费（时间就是金钱）。 而手工处理工作流程，一方面无法对整个流程状况进行有效跟踪、了解， 另一方面难免会出现人为的失误和时间上的延时导致效率低下，特别是无法进行量化统计，不利于查询、报表及绩效评估。

**三、Java开发者会为什么要学Activity工作流**

在Java领域，JBPM和Activity是两个主流的工作流系统， 而Activity的出现无疑将会取代JBPM（Activity的开发者就是从Jbpm开发者出来的）。

**四、Activity工作流学习要点**

1、1个插件 打开idea 选择 file-settings，打开如下页面，选择左侧Plugins ， 然后点击下面的Search…输入actiBPM搜索,安装并重启IDEA

2、1个引擎 ProcessEngine对象，Activity工作流引擎。这是Activiti工作的核心。 负责生成流程运行时的各种实例及数据、监控和管理流程的运行。 所有的操作都是从获取引擎开始的，所以一般会把引擎作为全局变量 ProcessEngine processEngine =ProcessEngines.getDefaultProcessEngine();1个配置文件 activiti.cfg.xml。Activiti核心配置文件， 配置流程引擎创建工具的基本参数和数据库连接池参数

3、5种数据库表

Activiti的后台是有数据库的支持，所有的表都以ACT\_开头。 第二部分是表示表的用途的两个字母标识。用途也和服务的API对应。

ACT\_RE\_\*: 'RE'表示repository。 这个前缀的表包含了流程定义和流程静态资源（图片，规则，等等）。

ACT\_RU\_\*: 'RU'表示runtime。 这些运行时的表，包含流程实例，任务，变量，异步任务，等运行中的数据。 Activiti只在流程实例执行过程中保存这些数据，在流程结束时就会删除这些记录。 这样运行时表可以一直很小速度很快。

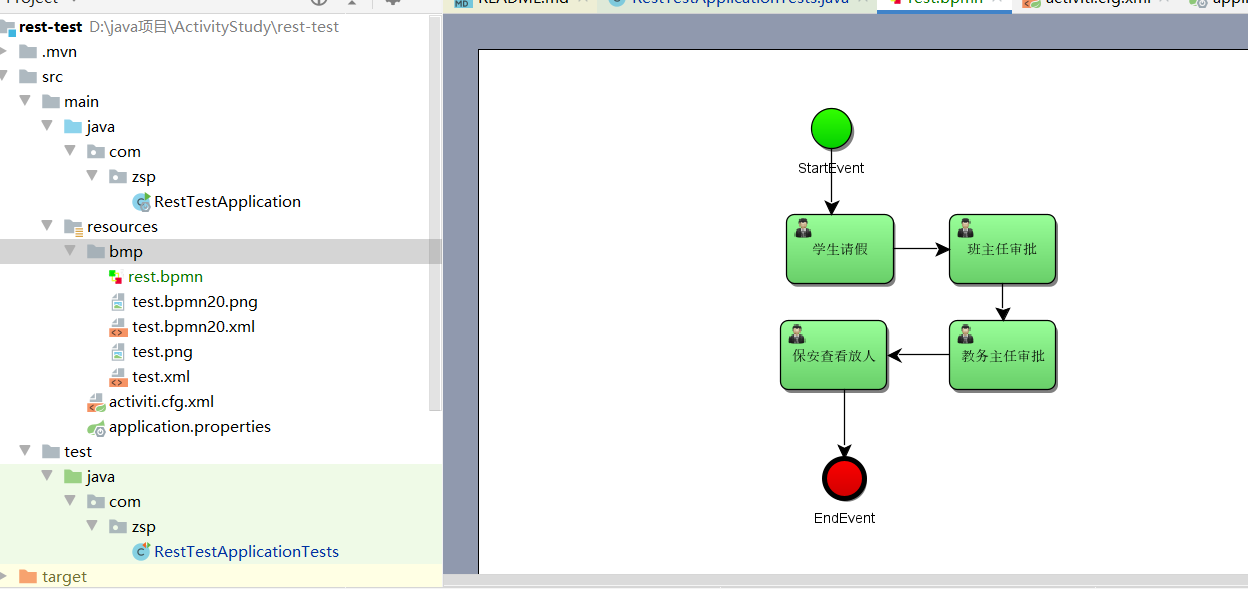
ACT\_ID\_\*: 'ID'表示identity。 这些表包含身份信息，比如用户，组等等。

ACT\_HI\_\*: 'HI'表示history。 这些表包含历史数据，比如历史流程实例，变量，任务等等。

ACT\_GE\_\*: 通用数据，用于不同场景下，如存放资源文件。

**使用流程图在activity工作流的情况下生成数据库表**

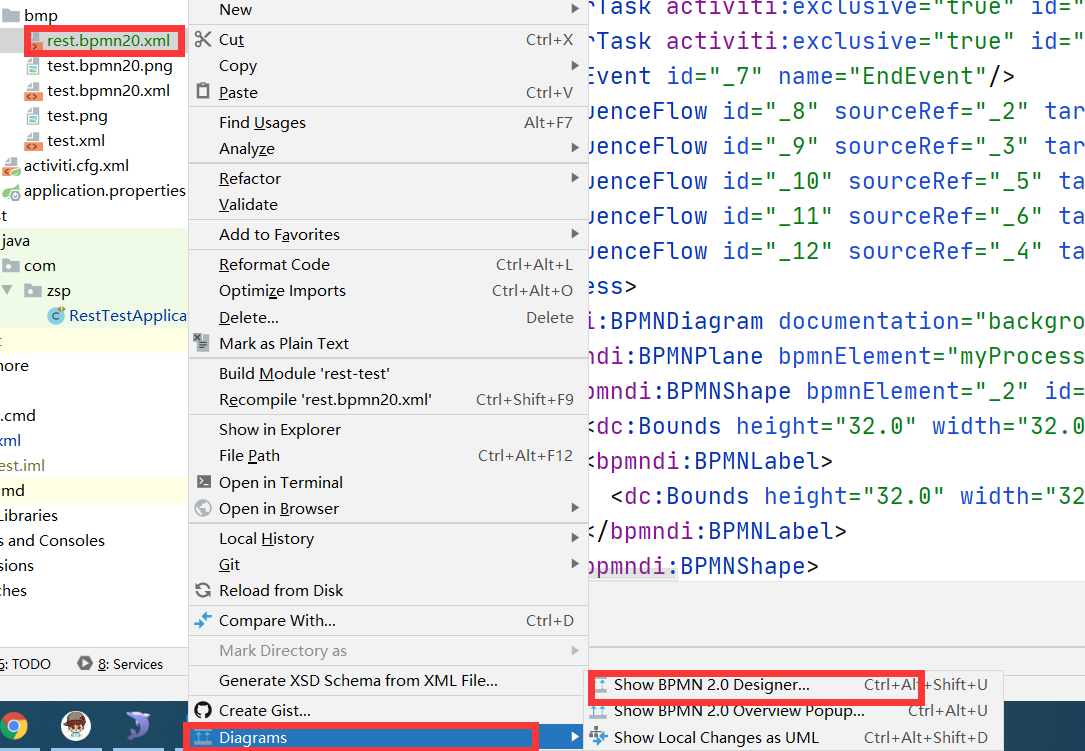
1. 先安装actiBPM插件，然后新建bmp流程图



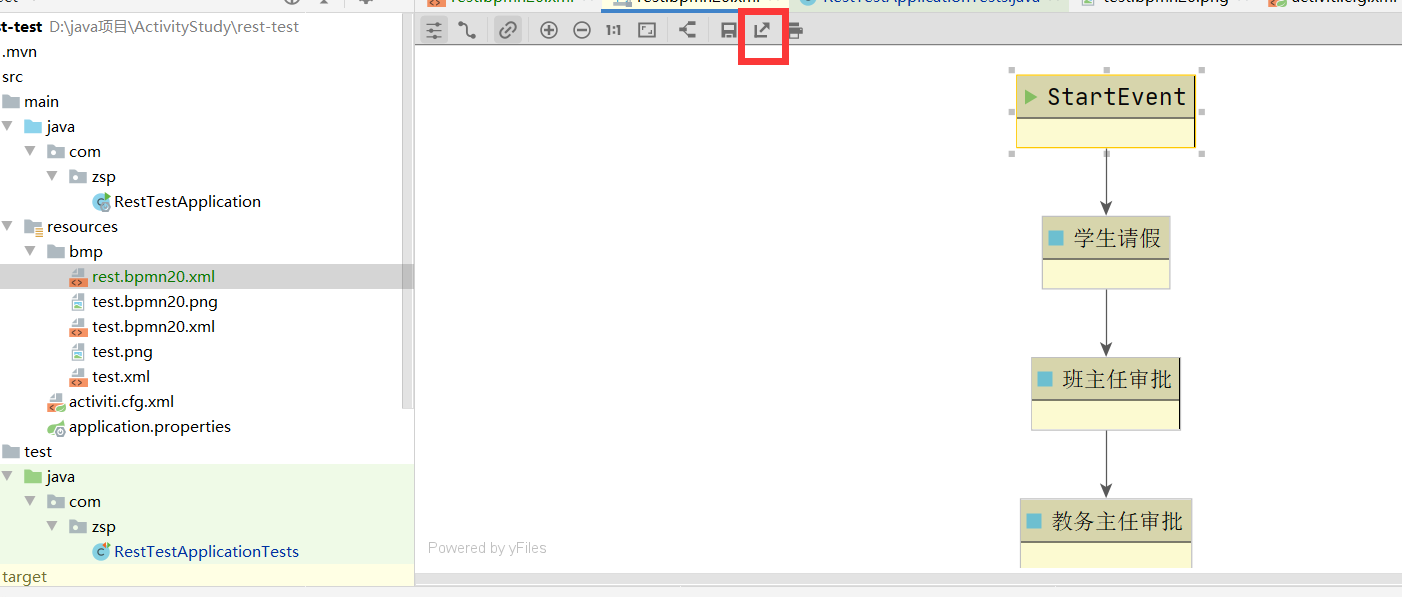
1. 将名字改成bpm20.xml后缀（这里一定要是bpm20否则act\_re\_procdef表不会插入数据）



1. 生成流程图



4、导出流程图

5、新建activiti.cfg.xml，这里特别注意高版本mysql的url一定要使用com.mysql.cj.jdbc.Driver还有nullCatalogMeansCurrent=true

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">  
 <bean id="processEngineConfiguration" class="org.activiti.engine.impl.cfg.StandaloneProcessEngineConfiguration">  
<!-- 配置数据库相关-->  
<!-- 数据库驱动-->  
 <property name="jdbcDriver" value="com.mysql.cj.jdbc.Driver"></property>  
<!-- 数据库链接-->  
 <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/test?useSSL=false&amp; useUnicode=true&amp; characterEncoding=utf8&amp; serverTimezone=Asia/Shanghai&amp; nullCatalogMeansCurrent=true"></property>  
 <property name="jdbcUsername" value="root"></property>  
 <property name="jdbcPassword" value="123456"></property>  
<!-- 数据库表在生成时的依赖-->  
 <property name="databaseSchemaUpdate" value="true"></property>  
 </bean>  
</beans>

6、使用测试类生成23张数据表（这里默认使用resources下的activiti.cfg.xml）

// 使用activity工作流先生成数据库表：23张  
 @Test  
 void testCreateTable(){  
// 使用xml生成activity工作流  
 Logger logger = LoggerFactory.getLogger(RestTestApplicationTests.class);  
// 获取activity提供的工具类  
// 获取这个默认engine时就会自动创建数据库表  
 ProcessEngine engine = ProcessEngines.getDefaultProcessEngine();  
 System.out.println(engine);  
 }

1. 使用流程表生成审批流程

// 1、部署流程  
 @Test  
 public void createActivityTask(){  
 //获取默认的流程引擎  
 ProcessEngine processEngine = ProcessEngines.getDefaultProcessEngine();  
  
 processEngine.getRepositoryService()//得到库服务  
 .createDeployment()//创建部署  
 .name("askRest")  
 .addClasspathResource("bmp/rest.bpmn20.xml") //导入流程图 !!!!一定要加bpmn20  
 .addClasspathResource("bmp/rest.bpmn20.png") //导入流程文件!!!!一定要加bpmn20  
 .deploy(); //开始部署  
 }

1. 启用流程图生成的审批流程

//2、启动流程实例  
@Test  
 public void testStartProcessInstance(){  
 ProcessEngine processEngine = ProcessEngines.getDefaultProcessEngine();  
 //act\_re\_procdef表里的id，如果没生成，去看上一步，真是哔了狗  
 processEngine.getRuntimeService()  
 .startProcessInstanceById("myProcess\_1:1:4");  
 }

1. 请假人发起申请

//3、请假人发出请假申请  
@Test  
 public void testAsk(){  
// 获取activity7的引擎  
 ProcessEngine processEngine = ProcessEngines.getDefaultProcessEngine();  
// 查看act\_ru\_task表，然后把id给上，发起请假申请  
 processEngine.getTaskService().complete("2505");  
 }

1. 老师查看审批流程

@Test  
public void queryTask(){  
// 获得流引擎  
 ProcessEngine engine = ProcessEngines.getDefaultProcessEngine();  
// 创造搜索，下面的teacher是我们在创建bpmn的时候写的assignee值  
 List<Task> student = engine.getTaskService().createTaskQuery()  
 .taskAssignee("teacher")//这里可以是老师，可以是学生，可以是任何人  
 .list();  
 for (Task task : student) {  
 System.out.println(task);  
 }  
}

1. 老师通过审批，后续的审批一样了，都是通过10步的查询（改下teacher成要查询的人）然后进行审批

@Test  
public void testFinishTask\_manager(){  
// 获取流引擎  
 ProcessEngine engine = ProcessEngines.getDefaultProcessEngine();  
// 这里的id是上面查询生成的请假申请id  
 engine.getTaskService()  
 .complete("7502"); //查看act\_ru\_task表，或者通过上面查询获得  
}

执行几次10和11的操作后，如图：

