

Flowable是什么

Flowable---定义：使用Java编写的轻量级业务流程引擎

工作流引擎：指workflow作为应用系统的一部分，并为之提供对各应用系统由决定作用的根据角色、分工和条件的不同决定信息传递路由、内容等级等核心解决方案，

工作流引擎功能包括流程的节点管理，流向管理、流程样例管理.....重要功能

jbm3 -> jbm4 -> activiti -> flowable

流程定义

主要包含两个文件： bpmn文件和流程图片

Flowable--功能： 1、部署BPMN2.0流程定义

2、创建流程定义的流程实例

3、查询/访问运行种(或历史)的流程实例及相关数据

Flowable--优点： 1、可以将JAR形式发布的Flowable库加入应用或服务，来嵌入引擎。

2、可以使用Flowable REST API进行HTTP调用

3、提供了可使用简单方便的UI的Flowable应用

Flowable是activiti的fork

activiti是一个著名的工作流引擎(一个工作审批流)。Activiti项目是一项新的基于Apache许可的开源BPM平台(业务流程管理)，从基础开始构建，旨在提供支持新的BPMN2.0标准，包括支持对象管理组(OMG),可以定义流程、执行流程并以不同方式对其实现运行。

Activiti提供的服务：

ProcessEngineConfiguration 可以从中构建流程引擎的配置信息；

创建processEngine配置对象cfg，使用buildProcessEngine()方法，创建processEngine对象。

```

ProcessEngineConfiguration cfg = new StandaloneProcessEngineConfiguration()
    .setJdbcUrl("jdbc:mysql://localhost:3306/test?
        useUnicode=true&characterEncoding=UTF-8&serverTimezone=Asia/Shanghai")
    .setJdbcUsername("root")
    .setJdbcPassword("yuzaixia95..")
    .setJdbcDriver("com.mysql.jdbc.Driver")

.cfg.setDatabaseSchemaUpdate(ProcessEngineConfiguration.DB_SCHEMA_UPDATE_TRUE);

// 2、用实例的配置创建 ProcessEngine对象
ProcessEngine processEngine = cfg.buildProcessEngine();

```

RepositoryService: Acticiti仓库服务类,

管理流程定义。(仓库中存储流程定义的两个文件: bpmn文件和流程图片)。

该服务提供对流程定义和部署仓库的访问的服务。提供一些静态信息(不会改变, 至少不会经常改变的信息)

可用于将流程定义部署到流程引擎。使用creatDeployment()[创建一个新的部署]--deploy()方法

```

RepositoryService repositoryService = processEngine.getRepositoryService();
Deployment deployment = repositoryService.createDeployment()
    .addClasspathResource("holiday-request.bpmn20.xml")
    .deploy();

// 4.通过RepositoryService创建的ProcessDefinitionQuery对象实现。
ProcessDefinition processDefinition =
repositoryService.createProcessDefinitionQuery()
    .deploymentId(deployment.getId())
    .singleResult();
System.out.println("Found process definition : " +
processDefinition.getName());

```

ProcessDefinition:一个对象结构, 表示由活动和过渡组成的可执行流程。业务流程通常使用图形编辑器创建, 这些图形编辑器以某种文件格式存储流程定义。这些文件可以添加到Deployment控件, eg:Business Archive(.bar)文件。在部署时, 引擎会将流程定义文件解析为此类的可执行实例。可用于启动一个流程实例查询流程定义是否在流程引擎。使用 repositoryService(processEngine).createProcessDefinitionQuery()[查询流程定义]

RuntimeService: Activiti流程执行服务类,

执行(启动/推进/删除...流程实例)管理

```
// 5.***** 启动实例
```

```
RuntimeService runtimeService = processEngine.getRuntimeService();

Map<String, Object> variables = new HashMap<String, Object>();
variables.put("employee", employee);
variables.put("nrOfHolidays", nrOfHolidays);
variables.put("description", description);
ProcessInstance processInstance =
    runtimeService.startProcessInstanceByKey("holidayRequest",
variables);
```

TaskService: Activiti任务服务类,

跟踪UserTasks, 可以使用此服务创建任务、声明并完成任务, 分配任务等

```
TaskService taskService = processEngine.getTaskService();
List<Task> tasks =
taskService.createTaskQuery().taskCandidateGroup("managers").list();
System.out.println("You have " + tasks.size() + " tasks:");
for (int i = 0; i < tasks.size(); i++) {
    System.out.println((i + 1) + ") " + tasks.get(i).getName());
}
```

createTaskQuery()--返回一个新的TaskQuery, 可用于动态查询任务。

HistoryService: Activiti查询历史信息类

提供访问任务和表单相关的服务。

提供正在进行或过去的流程实例的信息的服务。这与运行时信息有所不同, 因为该运行时信息在任何给定时刻仅包含实际运行时状态, 并且针对运行时过程执行性能进行了优化。历史记录信息经过优化, 易于查询, 并在永久性存储中永久保留。

```
HistoryService historyService = processEngine.getHistoryService();
List<HistoricActivityInstance> activities =
    historyService.createHistoricActivityInstanceQuery()
        .processInstanceId(processInstance.getId())
        .finished()
        .orderByHistoricActivityInstanceEndTime().asc()
        .list();
```

creatHistoricActivityInstanceQuery():创建一个新的程序化查询来搜索HistoricActivityInstances。

IdentityService: 组织机构, 管理用户和组

FormService: 可选, 用于定义开始表单和任务表单

