**Activit+Dubbo**

**项目说明文档**

（v0.1）

亚信联创科技（中国）有限公司

亚信联创科技（中国）有限公司

城市智能与运营事业部

二零一七年九月

**文档修改记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **日期** | **版本** | **修改内容说明** | **修改人** | **审阅人** | **审阅日期** |
| 1 | 2017.9.21 | 0.1 | 创建文档 | 吴启迪 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[1 功能概述 4](#_Toc493757852)

[2 运行环境示意图 4](#_Toc493757853)

[3 功能详细说明 4](#_Toc493757854)

[3.1 服务提供方配置文件 4](#_Toc493757855)

[3.2 服务使用方配置文件 5](#_Toc493757856)

[3.3 服务使用方maven说明 5](#_Toc493757857)

[3.4 服务使用方使用接口方式 6](#_Toc493757858)

[3.5 其他细节处理逻辑说明 6](#_Toc493757859)

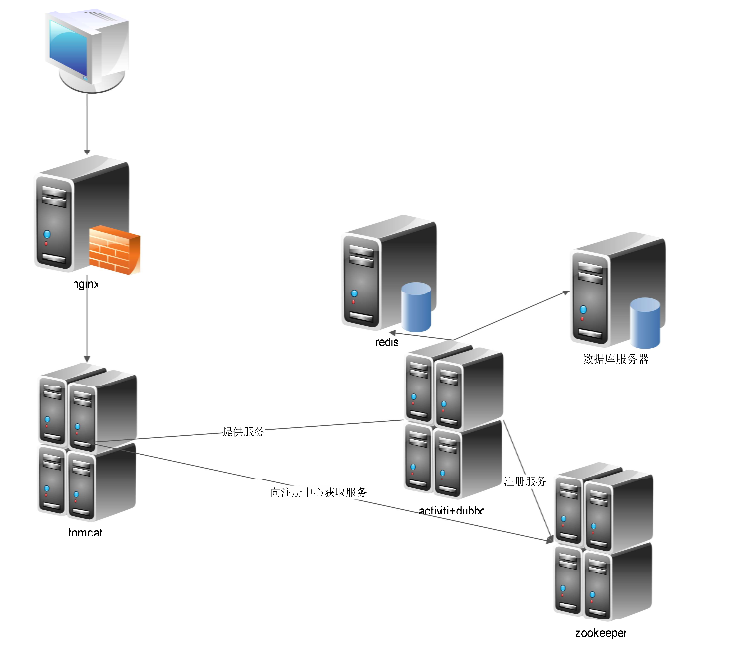
[3.6 根据测试进度待补充 6](#_Toc493757860)

**Activit+Dubbo项目说明**

# 功能概述

该项目主要功能是将工作流（activiti）的七大接口以分布式的形式对外提供服务，以满足各个项目对工作流服务的需求。

# 运行环境示意图



# 功能详细说明

## 服务提供方配置文件

<!-- 提供方应用信息，用于计算依赖关系 -->

<dubbo:application name=*"activiti-provider"* owner=*"wuqidi"*

organization=*"IOC"* />

<!-- 注册中心暴露服务地址 -->

<dubbo:registry group=*"dubbo"*

address=*"zookeeper://127.0.0.1:2181?client=zkclient"* check=*"false"* />

需要调整zookeeper地址 此处用于注册

<!-- 用dubbo协议在20880端口暴露服务 -->

<dubbo:protocol name=*"dubbo"* port=*"20880"* />

<dubbo:provider dispatcher=*"all"* timeout=*"50000"*

threadpool=*"fixed"* threads=*"500"* accepts=*"1000"* />

<!-- activiti对外开放的七大服务 -->

<dubbo:service interface=*"org.activiti.engine.RepositoryService"* ref=*"repositoryService"*/>

<dubbo:service interface=*"org.activiti.engine.RuntimeService"* ref=*"runtimeService"*/>

<dubbo:service interface=*"org.activiti.engine.FormService"* ref=*"formService"*/>

<dubbo:service interface=*"org.activiti.engine.IdentityService"* ref=*"identityService"*/>

<dubbo:service interface=*"org.activiti.engine.TaskService"* ref=*"taskService"*/>

<dubbo:service interface=*"org.activiti.engine.HistoryService"* ref=*"historyService"*/>

<dubbo:service interface=*"org.activiti.engine.ManagementService"* ref=*"managementService"*/>

<!-- ./activiti对外开放的七大服务 -->

## 服务使用方配置文件

<!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->

<dubbo:application name=*"demo-consumer"*/>

<!-- 使用注册中心暴露发现服务地址 -->

<dubbo:registry address=*"zookeeper://127.0.0.1:2181?client=zkclient"* />

需要调整zookeeper地址 此处用于注册

<!-- 生成远程服务代理，可以和本地bean一样使用demoService -->

<dubbo:reference interface=*"org.activiti.engine.ManagementService"* id=*"managementService"*/>

## 服务使用方maven说明

由于现在大部分项目没有使用maven，这里给出dubbo的maven，用maven命令自行下载添加到项目。

<parent>

<groupId>com.alibaba</groupId>

<artifactId>dubbo-demo</artifactId>

<version>2.5.5</version>

</parent>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>dubbo-demo-api</artifactId>

<version>${project.parent.version}</version>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>dubbo</artifactId>

<version>${project.parent.version}</version>

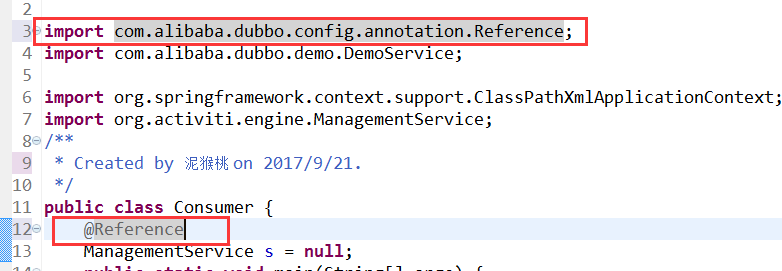
</dependency>

## 服务使用方使用接口方式

这里以*managementService*为例：

org.activiti.engine.ManagementService ser = (org.activiti.engine.ManagementService) context.getBean("managementService"); // 获取远程服务代理

或



## 其他细节处理逻辑说明

* 工作流的缓存使用redis实现。
* 现在项目中没有删除shiro功能，为以后提供其功能提供方便。
* 定时任务：使用quartz实现，为以后提供其功能提供方便。如过以后使用redis集群，可以把redis作为“中间库”，所有持久化操作直接操作redis，redis及时或定时更新到关系型数据库中。

## 根据测试进度待补充