# 流程引擎配置

[流程引擎配置 1](#_Toc23948)

[Java Config 1](#_Toc29549)

[XML Config 2](#_Toc17937)

[application.properties 2](#_Toc25982)

[ProcessEngineConfiguration抽象类及子类介绍： 3](#_Toc1177)

[ProcessEngine的三种创建方法 3](#_Toc9426)

两种方式：Java Config 和 XML Config

## Java Config

package com.example.demo.activiti.config;

import org.activiti.engine.ProcessEngine;

import org.activiti.engine.ProcessEngineConfiguration;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.annotation.Bean;

import org.springframework.core.env.Environment;

import org.activiti.spring.SpringProcessEngineConfiguration;

import org.activiti.engine.impl.cfg.StandaloneProcessEngineConfiguration;

import org.activiti.engine.impl.cfg.StandaloneInMemProcessEngineConfiguration;

@Configuration

public class ActivitiCfg {

@Autowired

private Environment environment;

@Bean

public ProcessEngineConfiguration processEngineConfiguration() {

ProcessEngineConfiguration cfg = new SpringProcessEngineConfiguration()

.setJdbcUrl(environment.getProperty("spring.datasource.url"))

.setJdbcUsername(environment.getProperty("spring.datasource.username"))

.setJdbcPassword(environment.getProperty("spring.datasource.password"))

.setJdbcDriver(environment.getProperty("spring.datasource.driverClassName"))

.setDatabaseSchemaUpdate(ProcessEngineConfiguration.DB\_SCHEMA\_UPDATE\_TRUE);

// ProcessEngine processEngine = cfg.buildProcessEngine();

// String pName = processEngine.getName();

// String ver = ProcessEngine.VERSION;

// System.out.println("==================>ProcessEngine [" + pName + "] Version: [" + ver + "]");

return cfg;

}

}

## XML Config

默认配置到src\main\resources\activiti.cfg.xml文件名固定！

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:context =*"http://www.springframework.org/schema/context"*

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*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

<!--加载配置文件-->

<context:property-placeholder location=*"classpath:application.properties"*/>

<!--流程引擎配置-->

<bean id=*"processEngineConfiguration"* class=*"org.activiti.engine.impl.cfg.StandaloneInMemProcessEngineConfiguration"*>

<property name=*"jdbcUrl"* value=*"${spring.datasource.url}"* />

<property name=*"jdbcDriver"* value=*"${spring.datasource.driverClassName}"* />

<property name=*"jdbcUsername"* value=*"${spring.datasource.username}"* />

<property name=*"jdbcPassword"* value=*"${spring.datasource.password}"* />

<property name=*"asyncExecutorActivate"* value=*"true"*/>

<property name=*"enableVerboseExecutionTreeLogging"* value=*"true"*/>

</bean>

<!-- 注入自动建表设置

<bean id="processEngine" class="org.activiti.spring.ProcessEngineFactoryBean">

<property name="processEngineConfiguration" ref="processEngineConfiguration"></property>

</bean>-->

</beans>

## application.properties

spring.datasource.url = jdbc:mysql://localhost:3306/demo?useUnicode=true&characterEncoding=UTF-8&useSSL=false

spring.datasource.username = root

spring.datasource.password = root

spring.datasource.driverClassName = com.mysql.jdbc.Driver

## ProcessEngineConfiguration抽象类及子类介绍：

* ProcessEngineConfiguration
  + ProcessEngineConfigurationImpl
    - StandaloneProcessEngineConfiguration
      * StandaloneInMemProcessEngineConfiguration
    - SpringProcessEngineConfiguration
* SpringProcessEngineConfiguration与Spring集成的流程引擎配置，可以用MYSQL。
* StandaloneProcessEngineConfiguration 单独的流程引擎配置。
* StandaloneInMemProcessEngineConfiguration H2数据库的In Memory的流程引擎配置

## [ProcessEngine的三种创建方法](https://www.cnblogs.com/lonecloud/p/5797532.html)

<https://www.cnblogs.com/lonecloud/p/5797532.html>