**集成activiti-modeler 5.22.0到项目中**

本文目的: 将activit 5.22.0 的 modeler 流程设计器 集成到自己的工程中去

解决问题:

1. 复制相关资源文件到自己的工程中

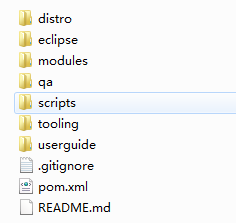
2. 解决modeler的路径访问问题,迁移到非系统根目录

3. 跟自己系统的spring无缝集成

首先请下载官方最新 5.22.0 发布包

下载地址: <https://github.com/Activiti/Activiti/archive/activiti-5.22.0.zip>

解压后如下图

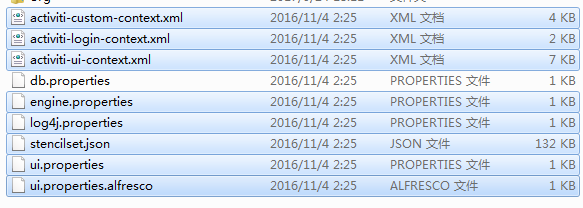


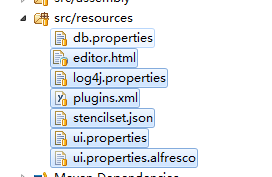
步骤:

1. 复制相关文件

1.1首先复制类路径资源文件

复制 \modules\activiti-webapp-explorer2\src\main\resources\下



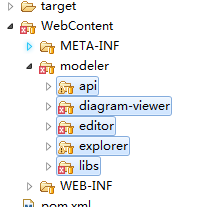


db.properties 没有选中  
   
 选中文件到你的classpath 根目录下,(src 或者resource 类路径下)

1.2 复制ui文件到你的webroot文件夹中

在 WebRoot目录中建立一个文件夹 取名 modeler

复制 modules\activiti-webapp-explorer2\src\main\webapp 下

  
   
 选中文件夹 到modeler目录下.

1.3 修改web.xml文件. 添加modeler的servlet

添加如下代码:

**Java代码  [收藏代码](javascript:void())**

1. <!-- Restlet adapter, used to expose modeler functionality through REST -->
2. <servlet>
3. <servlet-name>RestletServlet</servlet-name>
4. <servlet-**class**>org.restlet.ext.servlet.ServerServlet</servlet-**class**>
5. <init-param>
6. <!-- Application **class** name -->
7. <param-name>org.restlet.application</param-name>
8. <param-value>org.activiti.explorer.rest.application.ExplorerRestApplication</param-value>
9. </init-param>
10. </servlet>


14. <!-- Catch all service requests -->
15. <servlet-mapping>
16. <servlet-name>RestletServlet</servlet-name>
17. <url-pattern>/modeler/service/\*</url-pattern>
18. </servlet-mapping>

 首先确认. 你的工程引入了spring的监听. 也就代表你工程有spring .  因为activit-modeler严重依赖spring.

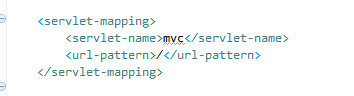
如果你工程没有spring环境, 建议下面就别看了. (先把自己的ssh , ssi , ssm 等框架搭建起来).

如果你使用的是spring mvc  建议你在你的mvc控制文件中加入

**Java代码  [收藏代码](javascript:void())**

1. <mvc:resources location="/modeler/" mapping="/modeler/\*\*" />

因为很有可能你根我一样过滤的是所有路径



1.4 复制modules\activiti-webapp-explorer2\src\main\java\org\activiti\explorer\rest\application下的

ExplorerRestApplication.java文件到你的classpath路径下(会提示报错. 必须的. 因为还没有引入相关jar依赖)

1.5 在你的pom.xml文件中引入如下代码

仓库:

**Java代码  [收藏代码](javascript:void())**

1. <repository>
2. <id>buzzmedia</id>
3. <url>http://maven.thebuzzmedia.com</url> <!-- ImageScalr -->
4. </repository>
5. <repository>
6. <id>activiti</id>
7. <name>Activiti</name>
8. <url>https://maven.alfresco.com/nexus/content/repositories/activiti/</url>
9. </repository>
10. <repository>
11. <id>Alfresco thirdparty</id>
12. <url>https://maven.alfresco.com/nexus/content/repositories/thirdparty/</url>
13. </repository>
14. <repository>
15. <id>activiti-third-party</id>
16. <name>Activiti third party</name>
17. <url>https://maven.alfresco.com/nexus/content/repositories/activiti-thirdparty/</url>
18. </repository>
19. <repository>
20. <id>maven-restlet</id>
21. <name>Public online Restlet repository</name>
22. <url>http://maven.restlet.org</url>
23. </repository>

 jar依赖

**Java代码  [收藏代码](javascript:void())**

1. <dependency>
2. <groupId>org.activiti</groupId>
3. <artifactId>activiti-modeler</artifactId>
4. <version>${activiti.version}</version>
5. </dependency>
6. <dependency>
7. <groupId>org.activiti</groupId>
8. <artifactId>activiti-explorer</artifactId>
9. <version>${activiti.version}</version>
10. <exclusions>
11. <exclusion>
12. <groupid>com.vaadin</groupid>
13. <artifactid>vaadin</artifactid>
14. </exclusion>
15. <exclusion>
16. <groupid>org.vaadin.addons</groupid>
17. <artifactid>dcharts-widget</artifactid>
18. </exclusion>
19. <exclusion>
20. <artifactid>activiti-simple-workflow</artifactid>
21. <groupid>org.activiti</groupid>
22. </exclusion>
23. </exclusions>
24. </dependency>
25. <!-- 查看流程详细定义 -->
26. <dependency>
27. <groupId>org.activiti</groupId>
28. <artifactId>activiti-diagram-rest</artifactId>
29. <version>${activiti.version}</version>
30. </dependency>

修改 diagram-viewer\index.html

文件.

如果你打算使用他的流程跟踪的话(建议修改,5.13版本跟踪页面已经支持IE了.不集成还费劲自己搞跟踪图提示呢?)

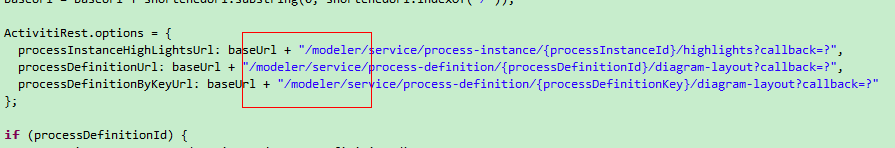
只需要使用5.13的

jstools.js

ProcessDiagramCanvas.js

在 \modules\activiti-webapp-explorer2\src\main\webapp\diagram-viewer\js 下

这两个js 替换 5.12.1的同名js即可



OK编译,clean 下载jar. ....

里面的spring版本替换成你自己的版本

OK了. 至此,集成到项目中也就完成了.

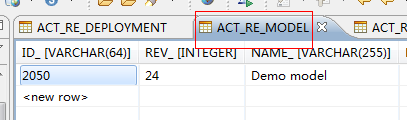
是否要测试下?

启动

访问:

[http://localhost:8080/YouPRJ/modeler/service/editor?id=2050](http://localhost:8080/Lams/modeler/service/editor?id=2050)

2050 替换成自己的流程部署文件id



OK可以直接编辑. 保存.

=======================================================================

以上是嵌入部署.

下面是独立部署. 你也可以直接使用官方的activiti-explorer  (呵呵.有点大.多余功能用不上)

这个是我从explorer里面分离出来的 单独modeler模块. (maven-eclipse工程)

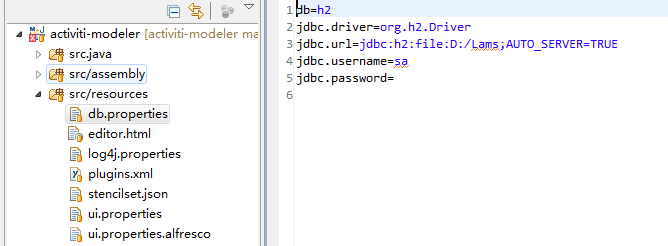
<https://github.com/izerui/activiti-modeler>

可以打包war包.跟你的项目放同一个web容器中.即可使用

[http://localhost:8080/activiti-modeler/modeler/service/editor?id=2050](http://localhost:8080/Lams/modeler/service/editor?id=2050)

直接设计保存到数据库.

哦忘了. 前提你要修改数据库连接.哈哈



OK 了.   至此   activiti-modeler 嵌入式部署. 和独立式部署 都完成了.

建议:

使用独立部署方式. 只是在你的web容器中多了一个activiti-modeler.war而已.

效果你可以使用iframe.因为在同域下. 其实跟嵌入式都一样的.