

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

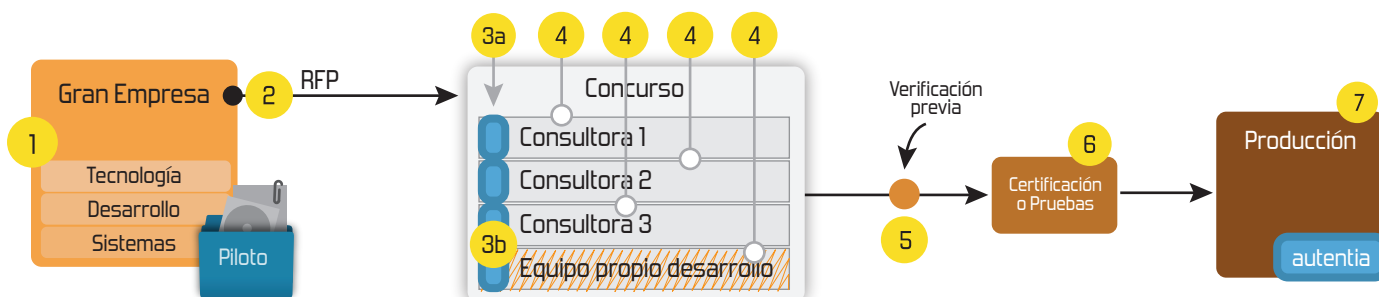
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD



BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)

AdictosAlTrabajo

Final de Terrakos
¡¡Ven al estreno!!
terrakos.com



autentia
Soporte a desarrollo informático
Hosting patrocinado por
enredados

Entra en Adictos a través de  

E-mail

Contraseña

Entrar Deseo registrarme
Olvidé mi contraseña

[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestros libros](#) [Más](#)

» Estás en: Inicio » Tutoriales » jBPM Form Builder: generación de formularios para jBPM5 y su integración en...



Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE



[Ver todos los tutoriales del autor](#)

Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2013-03-13

Tutorial visitado 1 veces [Descargar en PDF](#)

jBPM Form Builder: generación de formularios para jBPM5 y su integración en Guvnor.

0. Índice de contenidos.

- 1. Introducción.
- 2. Consideraciones previas.
- 3. Entorno.
- 4. Instalación.
- 5. Integración en Guvnor.
- 6. Creación de un formulario basado en la definición de una tarea humana.
- 7. Referencias.
- 8. Conclusiones.

1. Introducción

En este tutorial vamos a ver una pieza más del ecosistema BPMS de jBPM5, jBPM Form Builder, la pieza que nos va a permitir generar formularios para nuestras tareas de usuario dentro de un proceso BPMN2, mediante una interfaz visual tipo drag&drop.

jBPM Form Builder es una aplicación web más que desplegaremos sobre un Apache Tomcat 7, solo para diferenciarla del núcleo del motor de procesos que sí lo tenemos corriendo en un Jboss AS 7 y que se comunica con el repositorio de conocimiento, con Guvnor, haciendo uso del API REST de éste último. Gracias a dicha integración, vamos a poder:

- asociar un formulario a la ejecución de una tarea humana concreta de un flujo de trabajo, y
- generar un formulario de forma automática conforme a la definición de parámetros de entrada y salida de una tarea humana.

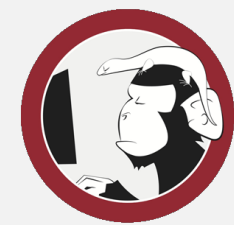
Lo más interesante de la suite es precisamente este componente, que nos va a permitir generar los formularios de cada una de las tareas particulares almacenándolos directamente en Guvnor, junto con el resto de "assets" (reglas de negocio, flujos de trabajo,...) relacionados con nuestros procesos de negocio.

jBPM Form Builder puede trabajar de dos formas distintas, como una aplicación web normal o integrada en Guvnor. En este tutorial vamos a analizar los pasos necesarios para configurar dicha integración y poder crear y modificar formularios directamente desde Guvnor.

2. Consideraciones previas.

Nos hemos encontrado con varios comportamientos anómalos, realizando las pruebas en el entorno en el que veníamos trabajando hasta ahora (jBPM 5.4.0.Final, Drools 5.5.0.Final y JDK 1.7.37 en MAC OS 10.7.5):

- si quisiéramos desplegar todo el ecosistema en Jboss 7, hay problemas de comunicaciones, de los que ya hemos hablado, y la causa es correr Jboss sobre la [versión 7 de la JDK](#) que nos ha obligado a bajar a la versión 1.6.0_37,
- errores de integración entre Guvnor y el Form Builder, que [nos han obligado a bajar a la versión 5.3.0.Final de jBPM](#) y a la versión 5.4.0-SNAPSHOT de Drools, con la versión 5.4.0-Final no están resueltos, véase:
 - en el Form Builder incrustado en Guvnor se muestran las opciones de "logout" y "guardar" cuando deberían estar ocultas, solo deberían mostrarse las del menú superior, el de Guvnor,
 - no se muestra la paleta de componentes de control, justo los interesantes para crear los formularios,



Síguenos a través de:



Últimas Noticias

» Vendedor: Soy inseguro, filtra o elige por mí: si quieres que te compre.

» [Comentando el libro: El arte de pensar, de Rolf Dobelli](#)

» [Ya está a la venta mi segundo libro: Planifica tu éxito, de aprendizaje a empresario](#)

» Ya esta disponible en eBook mi primer libro: Informática Profesional

» [Comentando el libro: La inteligencia reformada, las inteligencias múltiples en el siglo XXI de Howard Gardner](#)

[Histórico de noticias](#)

Últimos Tutoriales

» [Promesas: Organiza tu código Javascript/Coffeescript](#)

» [jBPM5 Console Server and Human Task Server: instalación y configuración](#)

» [Spring Security: haciendo](#)

- o no crea el fichero -taskform autogenerado que asocia el formulario personalizado con la tarea, con lo cuál no puedes hacer uso del mismo desde la consola,
- o no descubre los procesos hasta que no guardas la primera vez el formulario porque por defecto busca en el paquete defaultPackage (esto no es una peora de la versión 5.4, en la 5.3 tampoco funciona si tienes tu propia estructura de paquetes) y
- o al crear un flujo de trabajo no asigna el paquete seleccionado en el dialogo de creación y, del mismo modo, al crear un formulario no antepone el nombre del paquete al nombre del fichero; aunque termina funcionando no guarda coherencia con la estructura de paquete.
- nos hemos encontrado con [métodos por implementar](#) //fixme que dejan incompleta parte del API de comunicaciones con el motor; este punto no está aún solucionado, lo que nos obligaría en su caso a acceder directamente nosotros al LDAP para obtener esa información si la necesitásemos.

Con todo ello, hemos tenido que modificar el entorno de las pruebas al que enumeramos en el siguiente punto.

Un último apunte relacionado con el estado del proyecto lo podemos encontrar en los [recursos de la comunidad](#), más que en el [roadmap](#) que está desactualizado; el desarrollo de las aplicaciones web está basado en GWT, algunas con Seam3 y otras con el soporte de Spring y se están migrando a [UberFire](#), un framework para generar interfaces visuales basado en GWT, [Errai](#) y [CDI](#), que aún está en Alpha1; se está desarrollando en paralelo.

La versión 6 se basará en el core de todo este ecosistema, pero tendremos [nuevas implementaciones](#).

3. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.4 GHz Intel Core i7, 8GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Lion 10.7.4
- jBPM 5.3.0.Final
- Drools Guvnor 5.4.0-SNAPSHOT
- JDK 1.6.0_37
- Apache Tomcat 7.0.35

4. Instalación.

Sobre el proyecto en el que venimos trabajando, con el soporte de maven, deberíamos añadir un módulo más al pom.xml del proyecto `tnt-jbpm`, modificando las propiedades de las versiones como hemos comentado (también hemos renombrado los proyectos en relación a la versión del [tutorial anterior](#)):

```

1  <properties>
2    <java.version>1.6</java.version>
3    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
4    <drools.version>5.4.0-SNAPSHOT</drools.version>
5    <jbpm.version>5.3.0.Final</jbpm.version>
6  </properties>
7
8  ...
9
10 <modules>
11   <module>tnt-jbpm-designer</module>
12   <module>tnt-drools-guvnor</module>
13   <module>tnt-jbpm-gwt-form-builder</module>
14   ...
15 </modules>
```

El proyecto `tnt-jbpm-gwt-form-builder` tendría un `pom.xml` con el siguiente contenido:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema"
3    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4    <modelVersion>4.0.0</modelVersion>
5
6    <parent>
7      <groupId>com.autentia.bpm.repository</groupId>
8      <artifactId>tnt-jbpm</artifactId>
9      <version>0.0.1-SNAPSHOT</version>
10   </parent>
11
12   <groupId>com.autentia.bpm.repository</groupId>
13   <artifactId>tnt-jbpm-gwt-form-builder</artifactId>
14   <version>0.0.1-SNAPSHOT</version>
15   <packaging>war</packaging>
16
17   <inceptionYear>2013</inceptionYear>
18
19   <build>
20     <finalName>jbpm-form-builder</finalName>
21     <plugins>
22       <plugin>
23         <groupId>org.apache.maven.plugins</groupId>
24         <artifactId>maven-compiler-plugin</artifactId>
25         <version>2.5.1</version>
26         <configuration>
27           <source>${java.version}</source>
28           <target>${java.version}</target>
29         </configuration>
30       </plugin>
31       <plugin>
32         <groupId>org.apache.maven.plugins</groupId>
33         <artifactId>maven-surefire-plugin</artifactId>
34         <version>2.4.2</version>
35         <configuration>
36           <skipTests>true</skipTests>
```

uso de un servidor LDAP embebido.

» Introducción a Guvnor

» Gestión de expedientes en el ámbito de las Administraciones Públicas (IV): buscando en las forjas una solución.

Últimos Tutoriales del Autor

» jBPM5 Console Server and Human Task Server: instalación y configuración

» Spring Security: haciendo uso de un servidor LDAP embebido.

» Introducción a Guvnor

» Gestión de expedientes en el ámbito de las Administraciones Públicas (IV): buscando en las forjas una solución.

» Gestión de expedientes en el ámbito de las Administraciones Públicas (III): BPM y la gestión de procesos de negocio.

Últimas ofertas de empleo

2011-09-08
Comercial - Ventas - MADRID.

2011-09-03
Comercial - Ventas - VALENCIA.

2011-08-19
Comercial - Compras - ALICANTE.

2011-07-12
Otras Sin catalogar - MADRID.

2011-07-06
Otras Sin catalogar - LUGO.



José Manuel Sánchez
sanchezsuaarezj

sanchezsuaarezj @salaboy, integración de Form Builder en Guvnor con la v.5.4 de jBPM [kcy.me/guvv](#), te suena?, gracias! 21 hours ago · reply · retweet · favorite

adictosaltrabaj #jBPM5 Console Server and Human Task Server: instalación y configuración con el soporte de #maven - [kcy.me/gs3a](#) 2 days ago · reply · retweet · favorite

sanchezsuaarezj @aamormo muchas gracias hombre :) 8 days ago · reply · retweet · favorite

sanchezsuaarezj Spring Security: haciendo uso de un servidor LDAP embebido. - [kcy.me/gdem](#) 8 days ago · reply · retweet · favorite



Join the conversation

```
37         </configuration>
38     </plugin>
39     <plugin>
40         <groupId>org.apache.maven.plugins</groupId>
41         <artifactId>maven-war-plugin</artifactId>
42         <version>2.3</version>
43     </plugin>
44 </plugins>
45 </build>
46
47 <dependencies>
48
49     <dependency>
50         <groupId>org.jbpm</groupId>
51         <artifactId>jbpm-gwt-form-builder</artifactId>
52         <version>${jbpm.version}</version>
53         <type>war</type>
54     </dependency>
55
56 </dependencies>
57
58 </project>
```

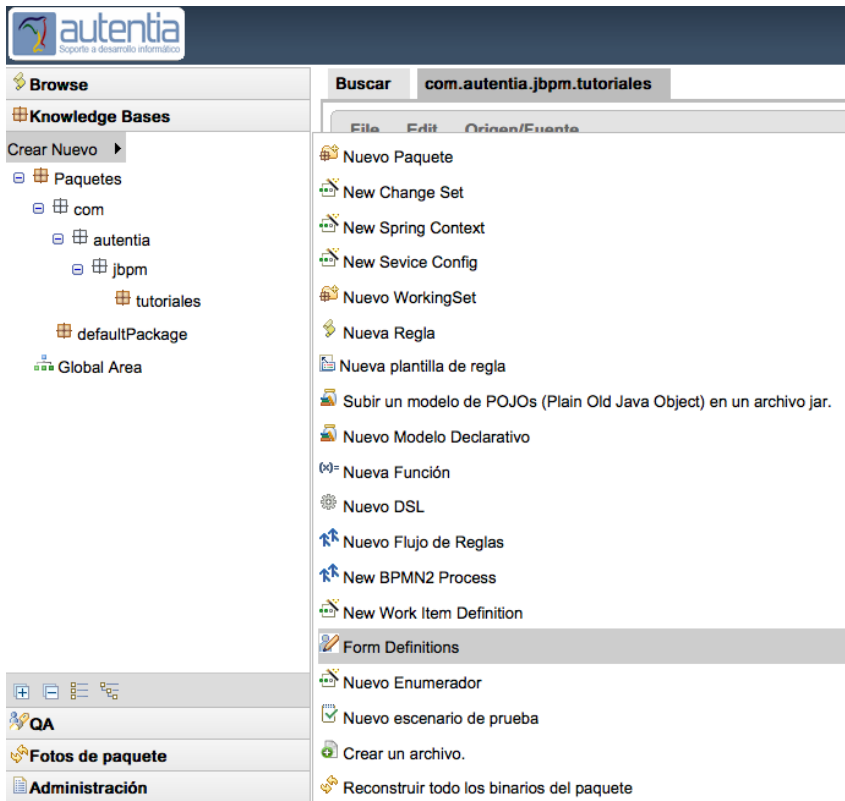
El proyecto no tiene más configuración que esta, lo que si debemos hacer es revisar la configuración de Guvnor, que contiene los parámetros de integración.

5. Integración en Guvnor.

Debemos añadir un fichero preferences.properties en el directorio src/main/resources del proyecto tnt-drools-guvnor, con el siguiente contenido:

```
1  #
2  # Global preferences for Guvnor web app.
3
4  visual-ruleflow=true
5  verifier=true
6  oryx-bpmn-editor=true
7  # asset.format.enabled.XYZ determines if format XYZ (as named inside guvnor extensions)
8  # is enabled on the current guvnor instalation
9  asset.format.enabled.formdef=true
10
11 # Date and language settings
12 drools.dateformat=dd-MMM-yyyy
13 drools.defaultlanguage=en
14 drools.defaultcountry=US
15
16 #RuleModeller (guided editor) checkbox default value
17 rule-modeller-onlyShowDSLStatements=false
18
19 #Designer configuration
20 designer.url=http://localhost:8080
21 #Do not change this unless you know what are you doing
22 designer.context=designer
23 designer.profile=jbpm
24
25 #FormBuilder configuration
26 #Do not change this unless you know what are you doing
27 formbuilder.url=http://localhost:8080
28 formbuilder.context=jbpm-form-builder
29 formbuilder.profile=guvnor
```

El contenido es muy similar al original, pero debemos asegurarnos de que las propiedades oryx-bpmn-editor y asset.format.enabled.formdef están activadas, porque en caso contrario el enlace que permite crear un formulario aparecería deshabilitado.



En este punto ya podríamos generar formularios desde Guvnor.

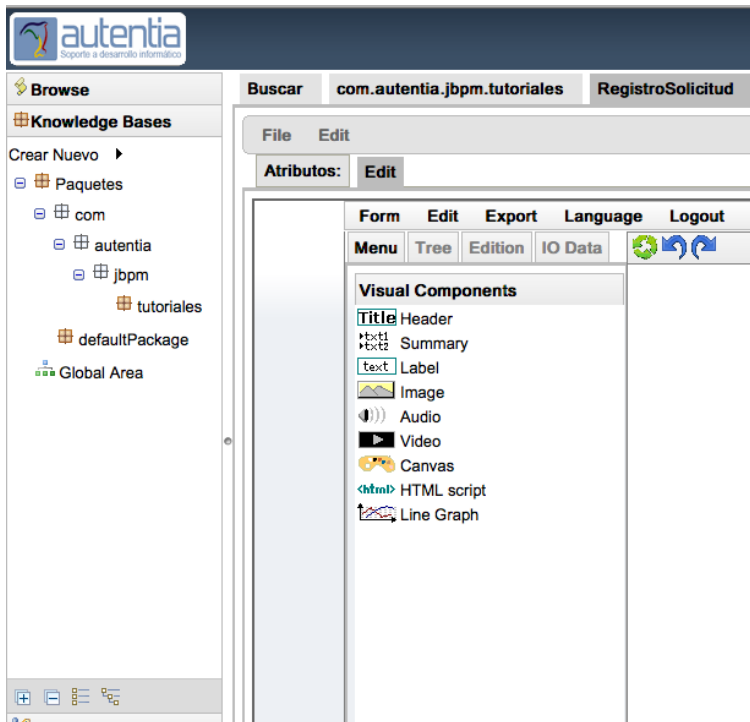
6. Creación de un formulario basado en la definición de una tarea humana.

Para crear un formulario basta con pulsar en "Crear Nuevo" > "Form Definitions" y se mostrará un diálogo como el siguiente:

Asignamos un nombre al formulario y tras pulsar ok, debería mostrarse la interfaz de usuario de creación del formulario, si bien antes nos solicita autenticación. Revisando la petición que se realiza y los parámetros que recibe el servlet que permite hacer uso de la interfaz embebida del Form Builder puede que haya un error en el envío o recepción de los parámetros de autenticación o esa parte está securizada cuando no debería serlo.

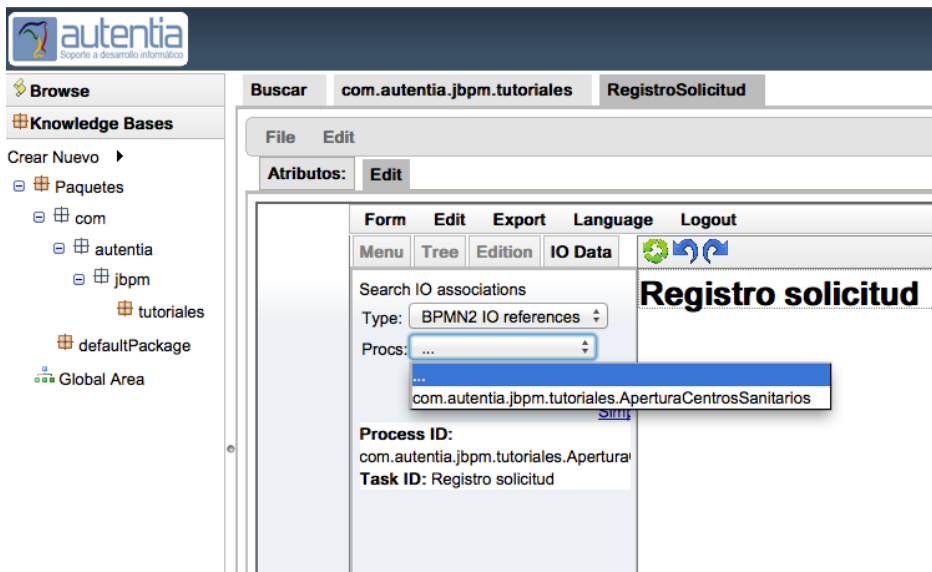
En cualquier caso, se mostrará un formulario de autenticación:

Tras incluir usuario y contraseña se mostrará la interfaz del Form Builder:



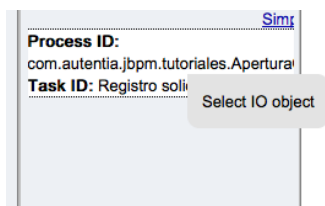
Como podéis comprobar, guarda coherencia porque se muestra una opción de logout; aunque creemos que no debería. Así mismo la opción de "Form", puesto que está en modo embebido, tampoco debería mostrarse.

Lo interesante es la pestaña "IO Data" que permite seleccionar un proceso de negocio y, de él, una tarea:



El problema en este punto es que, guardando una estructura de paquete, como la que venimos usando, debemos guardar primero el formulario para que se persista la información del paquete en el que estamos; por defecto busca en el paquete defaultPackage, pero tras guardarlo la primera vez sí se asignará. Además, no nos permite almacenar un formulario vacío, con lo que debemos añadir un título, por ejemplo, guardarlo y volver a abrirlo para que "funcione correctamente".

Sobre la tarea podemos pulsar "botón derecho" > "Select IO object" y el formulario se asignará automáticamente a la tarea del proceso correspondiente.



Una vez seleccionado podemos generar un formulario en base a los parámetros de entrada y salida pulsando de nuevo "botón

derecho" > "Quick Form from IO object"

Form Edit Export Language Logout

Menu Tree Edition IO Data

Process: com.autentia.jpdm.tutoriales.AperturaCentrosSanitarios

Search IO associations

Type: BPMN2 IO references

Procs: com.autentia.jpdm.tu

Search

Simr

Process ID:
com.autentia.jpdm.tutoriales.Apertura

Task ID: Registro solicitud

Inputs
TaskName
GroupId

Outputs
observaciones
documentos

Metadata
No metadata found

Quick Form from IO object

Se mostrará un diálogo como el siguiente desde el cuál podemos seleccionar los campos que nos interesen y el contenido del formulario se sobrescribirá con esta información:

Inputs to be added:

☒ TaskName

☒ GroupId

Outputs to be added:

☒ observaciones

☒ documentos

Warning: this will delete all the contents of your current form to create a simple form with all selected inputs and outputs. Are you sure you want to proceed?

Confirm Cancel

Se debería generar un formulario similar al siguiente:

Form Edit Export Language Logout

Menu Tree Edition IO Data

Process: com.autentia.jpdm.tutoriales.AperturaCentrosSanitarios

Search IO associations

Type: BPMN2 IO references

Procs: com.autentia.jpdm.tu

Search

Simr

Process ID:
com.autentia.jpdm.tutoriales.Apertura

Task ID: Registro solicitud

Inputs
TaskName
GroupId

Outputs
observaciones
documentos

Metadata
No metadata found

Task: Registro solicitud

Outputs:

observaciones

documentos

Complete

Tras pulsar "Guardar" o "Guardar y cerrar" en el listado de recursos deberíamos tener lo siguiente:

File Edit Origen/Fuente

Recurso Attribute Edit

Flujos de Reglas

Form Definitions

Refrescar lista Abrir seleccionado Open selected to single tab

Formato	Nombre	Estado	Ultimo modificado	Abrir
<input type="checkbox"/>	RegistroSolicitud	Draft	12/03/2013 11:46:14	Abrir

1 of 1

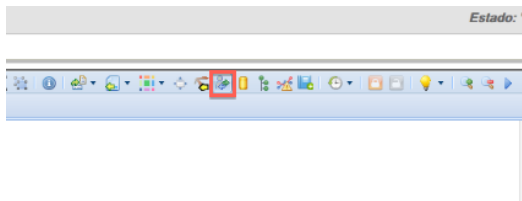
Otros recursos, documentación

Refrescar lista Abrir seleccionado Open selected to single tab

Formato	Nombre	Estado	Ultimo modificado	Abrir
<input type="checkbox"/>	Registro+solicitud-taskform automatic generation	Draft	12/03/2013 11:46:14	Abrir

Se crea un formulario autogenerado que apunta al formulario editable, y es en este punto donde tenemos el último error

relacionado con la ubicación de los recursos; parece que no se está anteponiendo el nombre del paquete, aunque no es seguro que sea ese el problema; la cuestión es que hay escenarios en las pruebas en las que no muestra el formulario desde la consola. Hasta que no generas los formularios de forma automática desde la interfaz del diseñador de procesos:



y vuelves a guardar el formulario desde el Form Builder, no se verá el formulario correctamente en la consola.

Si todo va bien, deberíamos ver algo como esto:

7. Referencias.

- <https://community.jboss.org/wiki/FormBuilderSetup>
- <http://vimeo.com/27344056>
- <http://blog.athico.com/2012/07/jbpm-form-builder-state-of-art.html>

8. Conclusiones.

Quizás originados por la migración que comentábamos o directamente porque el producto tiene bugs, como todos los productos, pero se hace difícil trabajar fuera del escenario de las demos que vemos y que tanto llaman nuestra atención. ¿Podríamos vivir sin el generador de formularios?, por supuesto, pero es lo interesante de la solución, ahorramos tener que generar manualmente las interfaces de cada una de las tareas del flujo desde nuestras aplicaciones, que son las que van a hacer uso del motor de estados.

Aún nos queda explorar cómo realizar la integración con sistemas externos desde el motor, creando nuestra propia librería de componentes de definición de tareas dentro de un flujo.

Como en ocasiones anteriores, os dejo [aquí](#) el proyecto mavenizado por si queréis jugar con él.

Stay tuned.

Un saludo.

Jose

jmsanchez@autentia.com

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)



Por favor, vota +1 o compártelo si te pareció interesante



Ánimate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Registrate** y accede a esta y otras ventajas «



SOME RIGHTS RESERVED

Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Copyright 2003-2013 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

W3C XHTML 1.0

W3C CSS

KML RSS

KML ATOM