

# ¿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

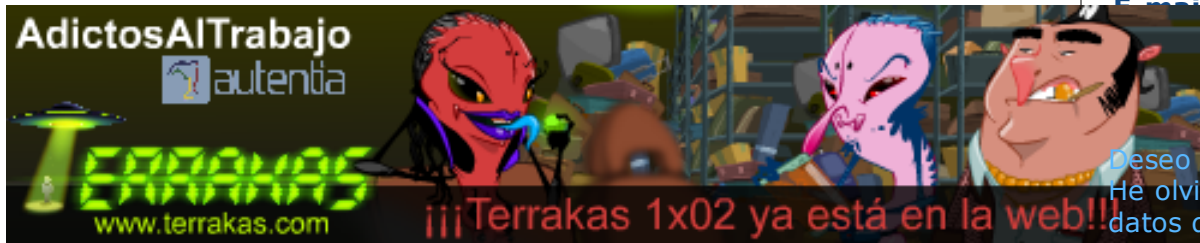
Tareas programadas (Quartz)  
Gestor documental (Alfresco)  
Inversión de control (Spring)

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)



Email:

Contraseña:

[Deseo registrarme](#)  
[He olvidado mis datos de acceso](#)

[Inicio](#) [Quiénes somos](#) [Tutoriales](#) [Formación](#) [Comparador de salarios](#)  
[Nuestro libro](#) [Charlas](#) [Más](#)

Estás en:

[Inicio](#) [Tutoriales](#) [Crear un proyecto de Mule ESB con Mule Studio](#)



**DESARROLLADO POR:**  
 [Juan Alonso Ramos](#)

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática de Gestión e Ingeniero en Informática, especialidad en Ingeniería del Software

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

Somos expertos en Java/J2EE

[Ver tutoriales de Juan Alonso Ramos](#)

Catálogo de servicios  
Autentia



**Fecha de publicación del tutorial: 2012-01-19**



Share |

0

[Regístrate para votar](#)

# Crear un proyecto de Mule ESB con Mule Studio


## Índice de contenidos.


- 1. Introducción
- 2. Entorno
- 3. Creación del proyecto Mule Studio
- 4. Conclusiones


## 1. Introducción


En este tutorial vamos a crear un proyecto de [Mule ESB](#) con la herramienta

## Últimas Noticias

 Cuando en alumno está preparado el maestro aparece

 [iii Los Reyes nos han traído Terrakas 1x02 !!!](#)

 Caso de éxito Tagus y cómo trabaja Autentia

 Todo preparado para el estreno de Terrakas 1x02

 [iiiConfirmado el jamón para el](#)

Mule Studio. Esta herramienta nos permitirá crear flujos de mensajes y servicios de Mule de forma gráfica lo cual es una gran ventaja para empezar a trabajar con este ESB ya que en ocasiones lo más complicado es la configuración del fichero mule-config.xml y este IDE nos facilita esta tarea.

## 2. Entorno

- MacBook Pro 15' (2.4 GHz Intel Core i5, 4GB DDR3 SDRAM).
- Sistema Operativo: Mac OS X Snow Leopard 10.6.8
- JDK 1.6.0\_29
- Mule 3.2.0

## 3. Instalación de Mule Studio

Mule Studio se distribuye como un Eclipse IDE modificado. Está disponible para Windows, Mac OS X 10.6 y Linux. La instalación es sencilla, descargarlo de la [página oficial](#) y descomprimir en un directorio.

[estreno del segundo capítulo de Terrakas!!!](#)



Histórico de NOTICIAS

Últimos Tutoriales



WeTransfer - Envío simple de archivos



Cómo utilizar la herramienta "huesos" de Adobe Flash para hacer andar a Álex (Terrakas).



Creación de una anotación de validación personalizada para Bean Validator.



Crear un proyecto de Mule ESB con Maven



Primeros pasos con Mule ESB

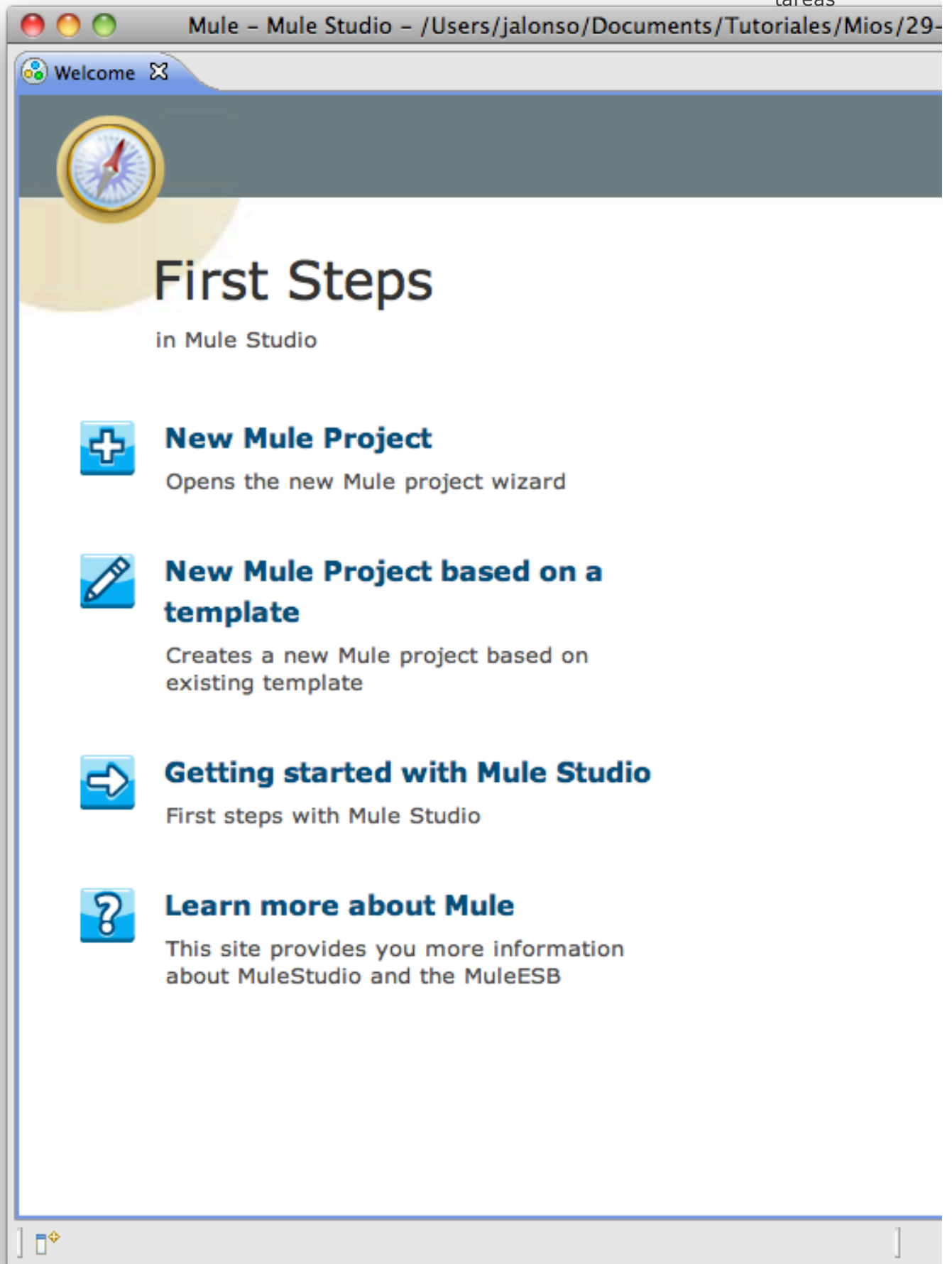
Últimos Tutoriales del Autor



Crear un proyecto de Mule ESB con Maven



Primeros pasos con Mule ESB



## 4. Crear un proyecto con Mule Studio

Una vez instalado Mule Studio ya sólo queda ponernos manos a la obra y empezar con nuestro primer proyecto Mule desde la opción New Mule Project. Le damos un nombre y una descripción a nuestro proyecto que consistirá en una aplicación encargada de recoger mensajes que le llegan vía Ajax, convertirlos a String y guardarlos en ficheros del disco.

Comercial -  
Compras -  
ALICANTE.

2011-07-12

Otras Sin  
catalogar -  
MADRID.

2011-07-06

**New Mule Project**

**Mule Project**  
Create a new Mule project.

**Project Documentation:**

Name: Messages register

Description: Aplicación que se encarga de recoger mensajes y guardarlos en un fichero.

**Server Settings:**

Server Runtime: Mule Server 3.2.1 CE

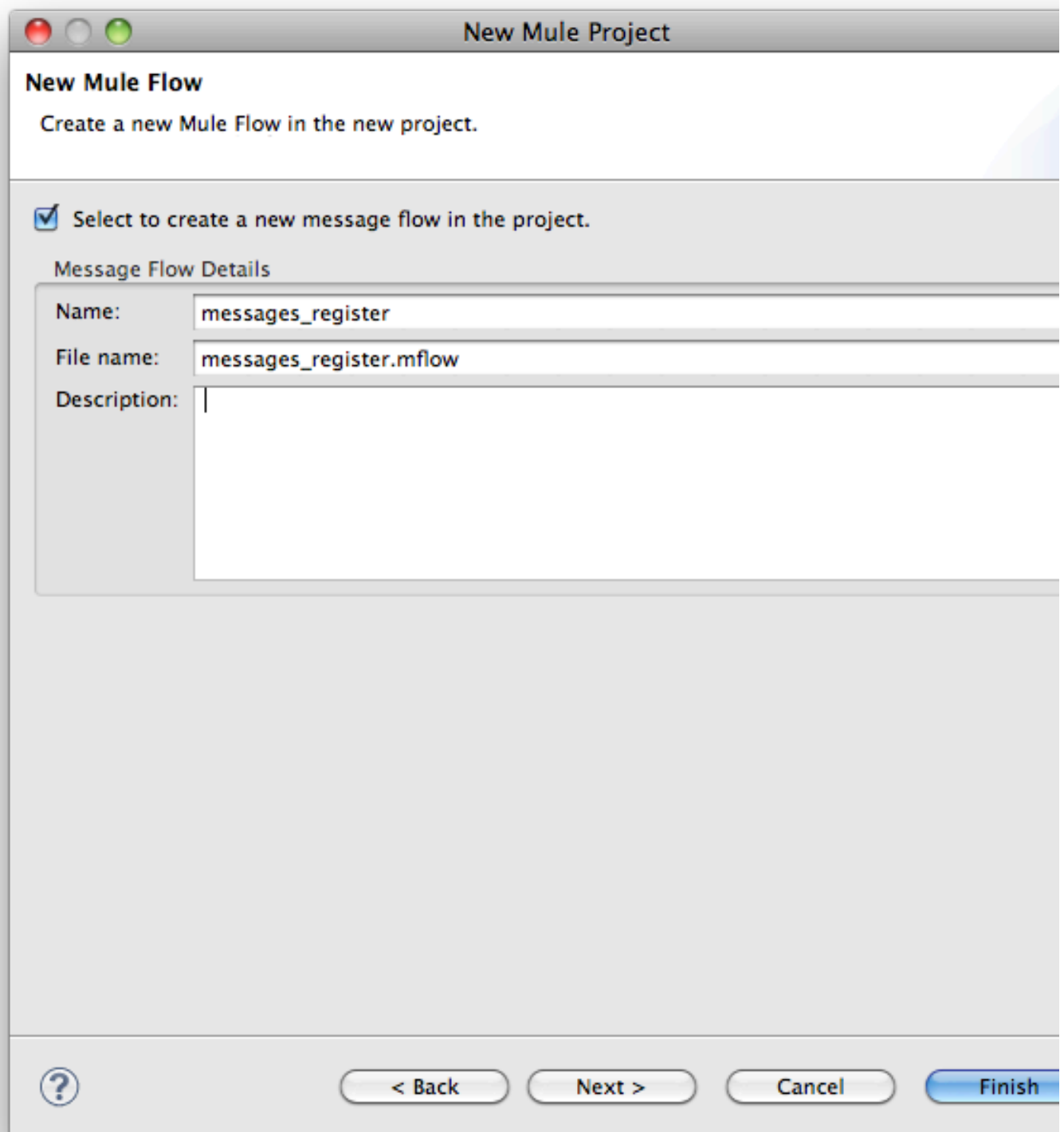
**Templates:**

☐ Create project based on an existing template.

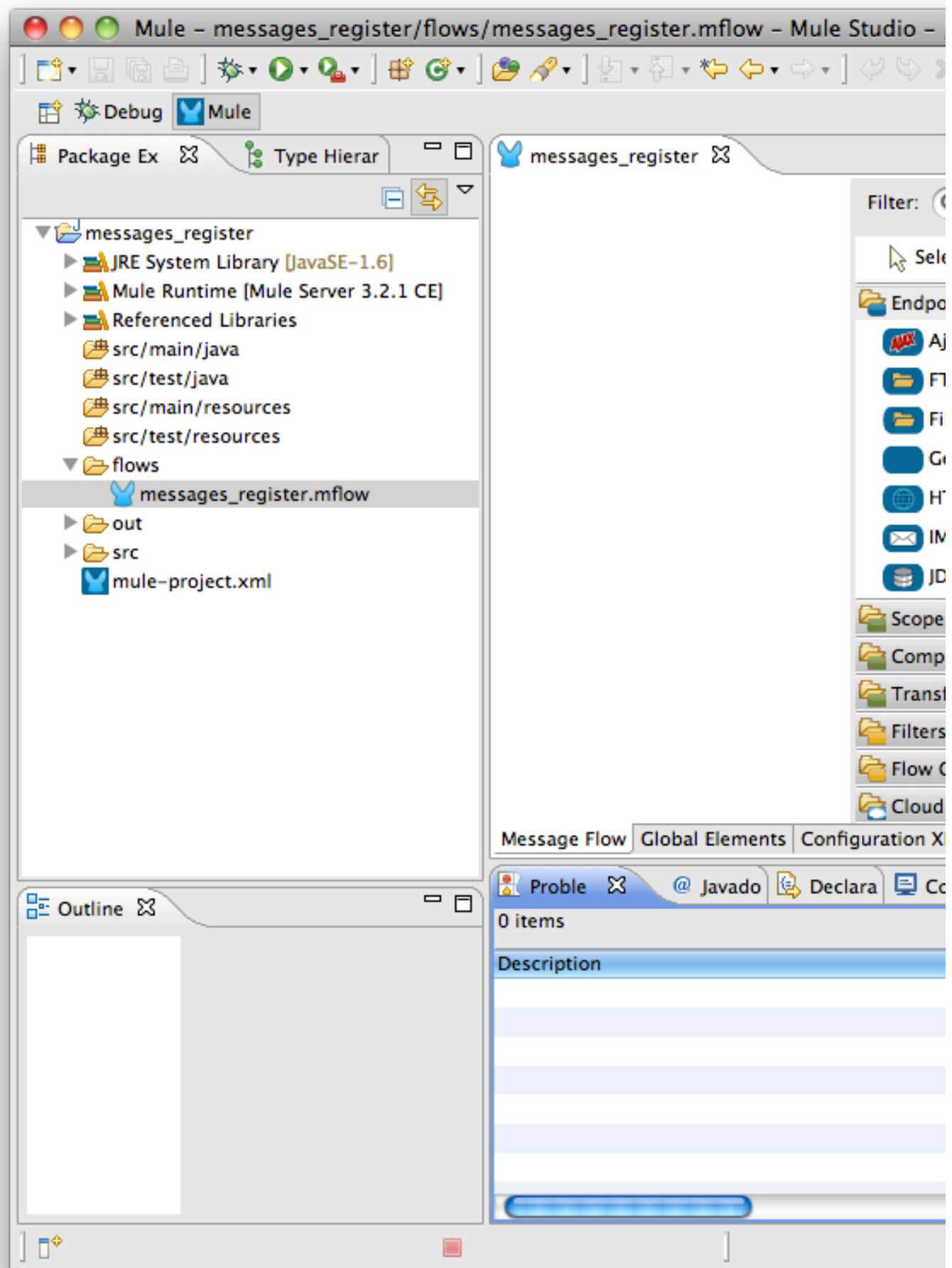
Name	Description
Echo	Echo example included in Mule distribution.
Order Discounter	Processes XML orders, applies tiered discounts, and adds Twitter status u
Order Fulfillment	Process XML orders, split into items and route for fulfillment.

? < Back Next > Cancel Finish

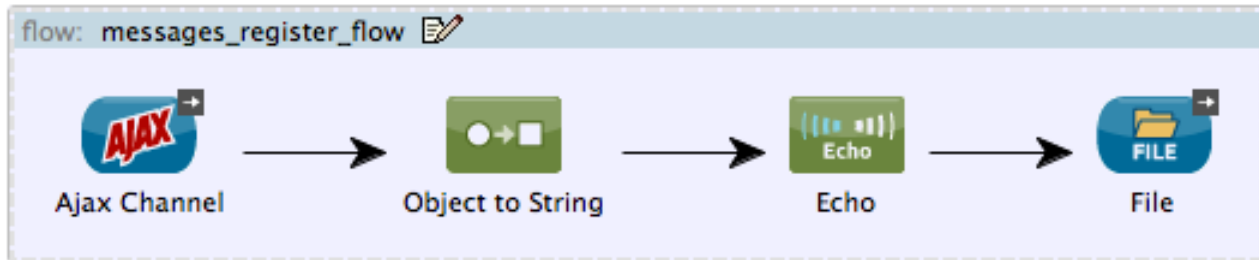
En la siguiente pantalla nos pide el nombre para el fichero que servirá para dibujar el flujo de mensajes con la herramienta gráfica.



Una vez creado el proyecto vemos que nos aparece a la derecha una paleta con todos los componentes que podemos añadir a un flujo agrupadas por categorías: endpoints, scopes, componentes, transformers, filters, flow control y cloud connectors.



El flujo contendrá un endpoint de ajax que recoge el objeto que le llega desde una petición, se lo pasa a un transformador que convierte el objeto ajax a string y a su vez se lo pasa a otro endpoint que se encarga de guardarlo en disco con un determinado formato. Entre medias también meteremos un log para ver el contenido de las peticiones. Para haceros una idea del contenido final del flujo podéis en la siguiente captura. Cada elemento lo iremos explicando con detalle.



Si vemos el contenido del XML generado podemos hacernos un poco la idea de lo que se encarga este IDE. La principal ventaja es que no tendremos que escribir nosotros el fichero de configuración de Mule sino que la herramienta lo hará por nosotros.

```

<?xml version="1.0" encoding="UTF-8"?>
<mule xmlns="http://www.mulesoft.org/schema/mule/core"
      xmlns:ajax="http://www.mulesoft.org/schema/mule/ajax"
      xmlns:file="http://www.mulesoft.org/schema/mule/file"
      xmlns:doc="http://www.mulesoft.org/schema/mule/documentation"
      xmlns:core="http://www.mulesoft.org/schema/mule/core"
      xmlns:http="http://www.mulesoft.org/schema/mule/http"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="CE-3.2"
      xsi:schemaLocation="
        http://www.mulesoft.org/schema/mule/ajax http://www.mulesoft.
        http://www.mulesoft.org/schema/mule/file http://www.mulesoft.
        http://www.mulesoft.org/schema/mule/core http://www.mulesoft.
        http://www.mulesoft.org/schema/mule/http http://www.mulesoft.

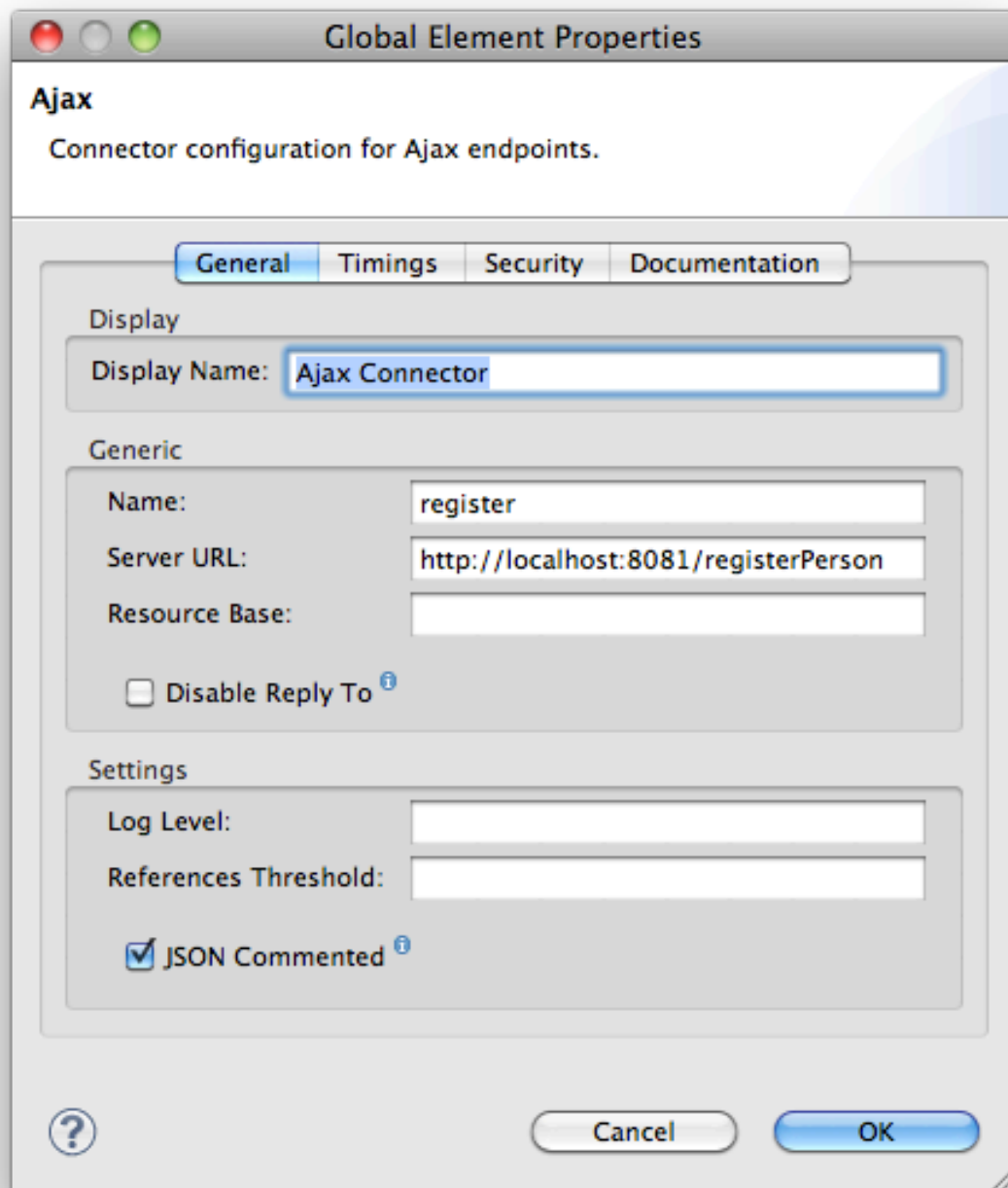
    <ajax:connector name="register" serverUrl="http://localhost:8081/registerPerson"
                    jsonCommented="true" doc:name="Ajax Connector" dis

    <flow name="messages_register_flow" doc:name="messages_register_flow">
      <ajax:inbound-endpoint channel="/register/message" connector-ref="register" dc
    <object-to-string-transformer doc:name="Object to String"/>
    <echo-component doc:name="Echo"/>
    <file:outbound-endpoint
      path="/tmp/workspace/messages_register/out"
      outputPattern="#[function:datestamp:dd-MM-yy]_[function:syst

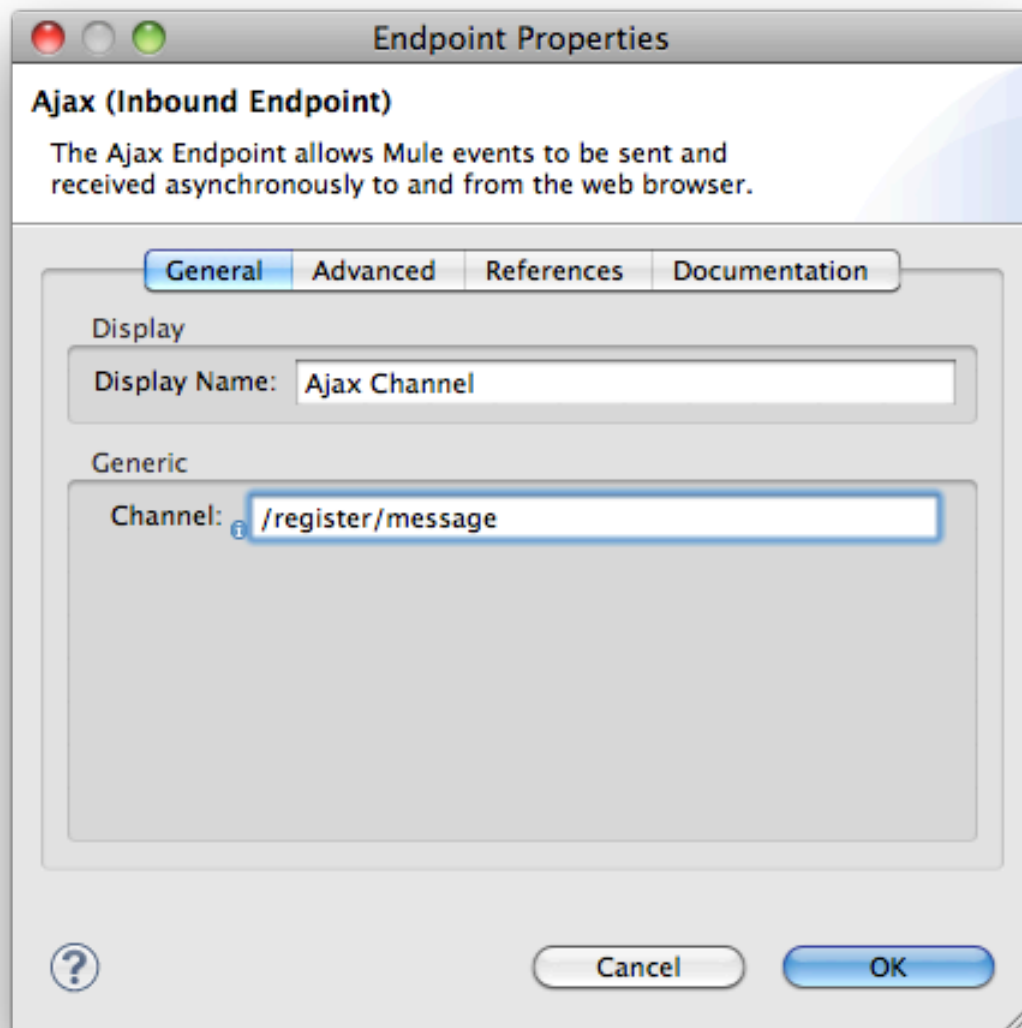
    </flow>
  </mule>
  
```

El primer elemento es un endPoint que recoge las peticiones HTTP servidas mediante AJAX a la dirección que nosotros le digamos. Para ello tenemos que añadir un conector al flujo de Mule desde la opción 'Global Elements' dentro de vista que muestra nuestro flujo. En la pantalla debemos añadir las propiedades genéricas para configurar el nombre del conector y la URL donde estará escuchando el mismo.

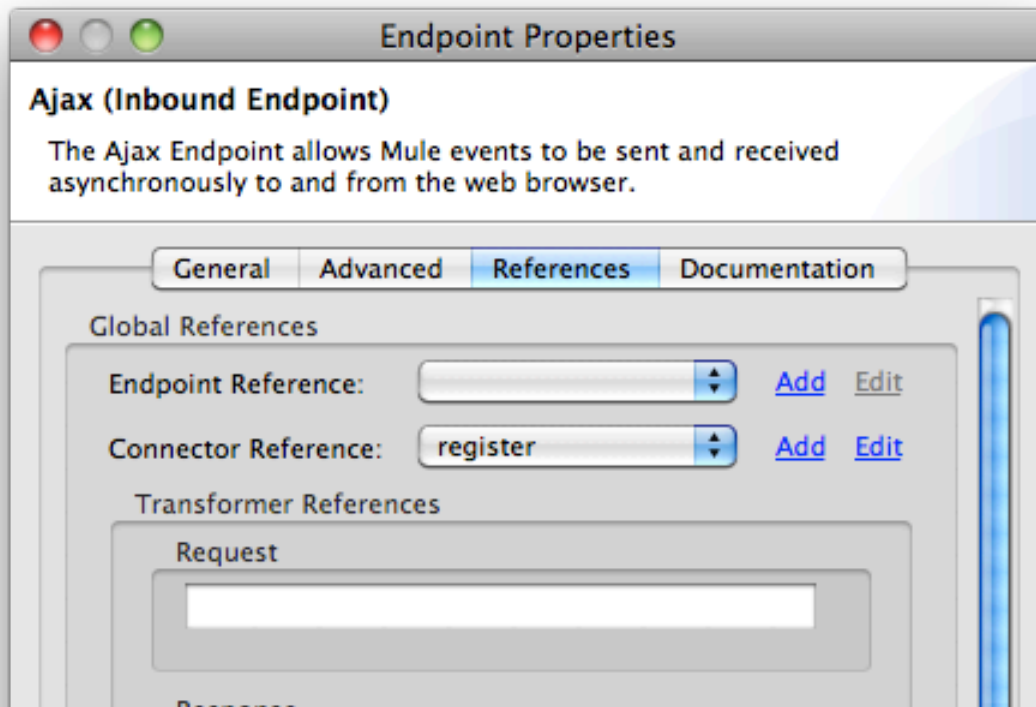




Haciendo doble clic en el elemento de Ajax del flujo se nos abre la ventana de propiedades donde es obligatorio indicar el canal para el endpoint donde atiende las peticiones, en nuestro ejemplo en /register/message.



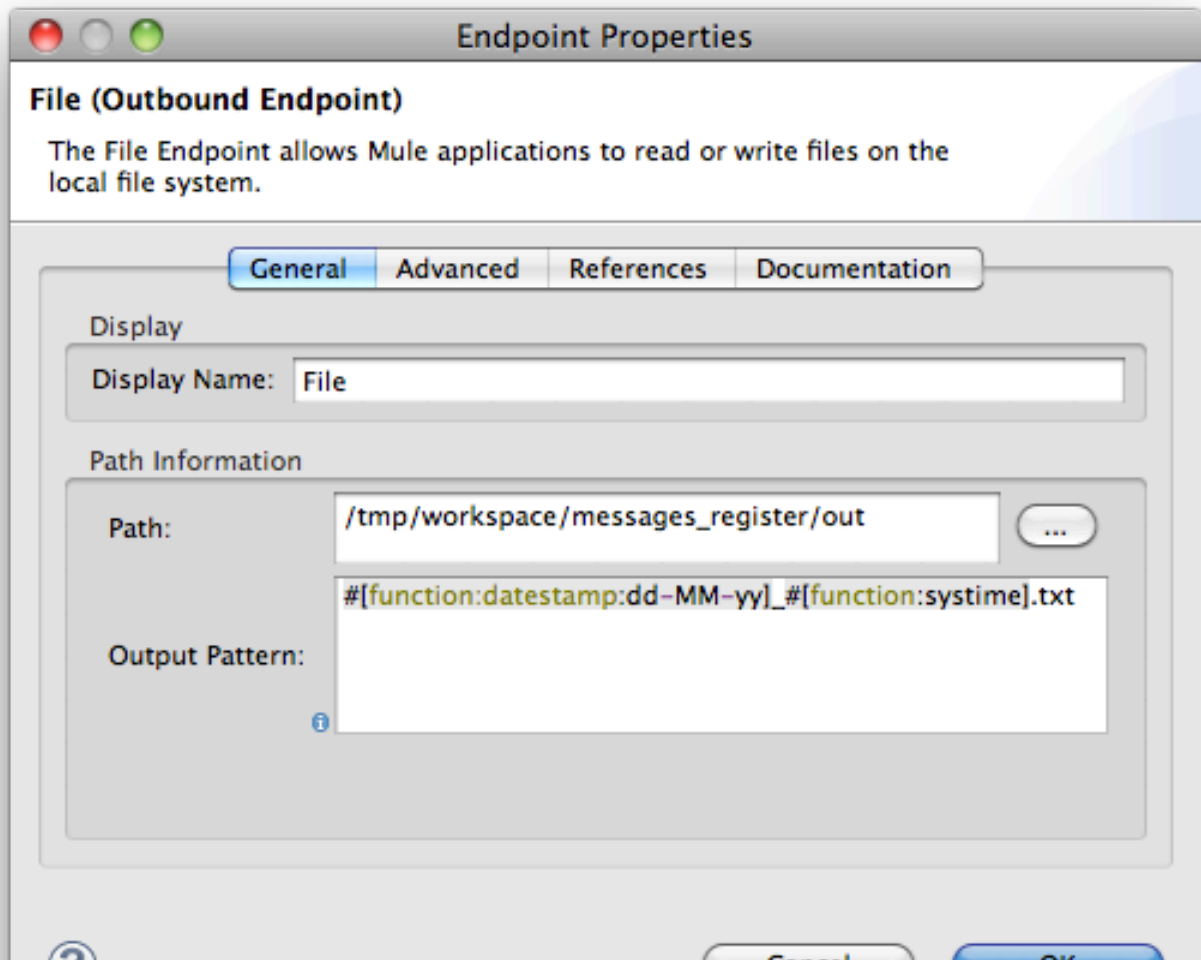
En la misma ventana en la pestaña References tenemos que hacer referencia al conector que configuramos anteriormente llamado 'register' que nos aparecerá en el desplegable.



A continuación el objeto Ajax queremos procesarlo y sacar el mensaje que lleva dentro, es decir la cadena de texto que envía el usuario junto con la petición. Para ello añadimos el transformador de objeto a string. Mule automáticamente mediante este transformador por defecto determinará el tipo de objeto que le llega y lo convertirá a String.

Para poder ir trazando las peticiones y ver que es lo que está haciendo el flujo de mensajes añadimos también un componente de log que es el componente 'Echo' que nos irá sacando por consola el contenido del payload resultante de la petición.

Por último el mensaje será almacenado en un fichero del disco mediante el endPoint de fichero donde le indicamos el path del directorio donde almacenar los mensajes y un formato para el nombrado del mismo.



Para probar el ejemplo debemos hacer una página HTML con un formulario donde un usuario pueda escribir mensajes que serán enviados al servidor vía AJAX donde serán recogidos por el endPoint configurado y redirigidos a una salida en disco.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script type="text/javascript" src="mule-resource/js/mule.js"></script>
<title>Guardar mensajes en disco del servidor</title>
<script type="text/javascript">
    function register() {
        mule.rpc("/register/message", document.getElementById('message').valu
    }

    function callBack(message) {
        document.getElementById("response").innerHTML = "Respuesta: " + message.d
    }
</script>

</head>
<body>
```

```
<fieldset>
    <legend>Guardar mensajes en disco del servidor</legend>
    Introduce un texto: <input type="text" id="message" />
    <input type="button" onclick="register()" value="Enviar"/>

</fieldset>
</div id="response"></div>
</body>
</html>
```

En la función 'register' hacemos una petición a través del API Javascript de Mule para hacer llamadas RPC, por eso debemos incluir el fichero Javascript `mule-resource/js/mule.js` que está disponible dentro del servidor stand-alone de Mule y estará disponible una vez que despluguemos nuestra aplicación.

La función `mule.rpc` recibe como primer argumento el canal por el cual se va a realizar la petición, el segundo es el objeto que deseamos enviar, en este caso el mensaje que introduce el usuario y el último argumento define una función de callback que será la que procese la respuesta.

La página resultante después de enviar una petición queda así:

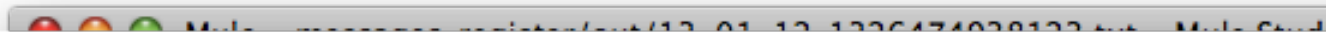


**Guardar mensajes en disco del servidor**

Introduce un texto:

**Respuesta: hola Mule!!**

Para terminar el ejemplo, en el disco de salida comprobamos que están llegando los mensajes y se está creando un fichero por cada uno de ellos.



## 4.Conclusiones

El ejemplo que hemos visto en este tutorial realizado de forma gráfica con Mule Studio no es un ejemplo demasiado complicado. Si repasamos el contenido del ejemplo podemos ver que hay una aplicación web corriendo en un servidor, a través de un cliente HTML con un formulario que mediante AJAX realiza peticiones a un servidor que las recoge, devuelve una respuesta mostrando el contenido del mensaje recibido y a su vez almacena

las peticiones en ficheros del disco.

Esto que no es complejo seguro que nos haría falta escribir mucho código fuente y como habéis visto, únicamente hemos escrito el código del cliente, unas 20 líneas de HTML y Javascript, todo lo demás nos lo ha proporcionado Mule. Con esto podemos ver la potencia de esta herramienta donde con muy poco código podemos comunicar diferentes servicios y sobre todo, hacerlos transparentes unos de otros.

Espero que te haya servido de ayuda.

Un saludo. Juan.

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

Share |

0

**También puedes seguirnos a través de:**



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

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

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

## COMENTARIOS



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-](#)

No comercial-Sin obras derivadas 2.5

Copyright 2003-2012. All Rights Reserved. All text is legal. Condiciones de uso |

