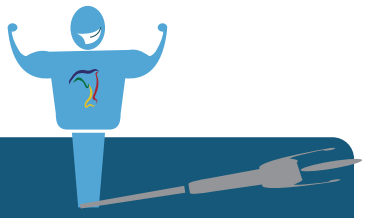
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

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

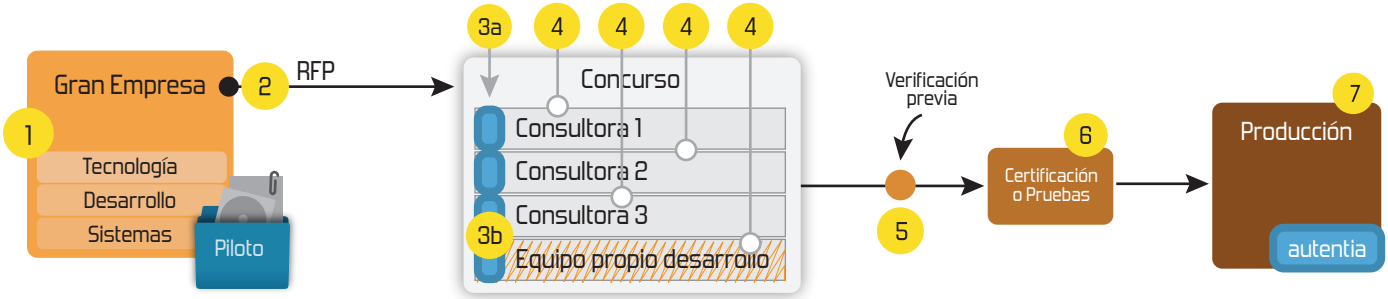
1. Desarrollo de componentes y proyectos a medida



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

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

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

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

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com



autentia
Soporte a desarrollo informático
Hosting patrocinado por enREDados

E-mail:
Contraseña:
Entrar
Deseo registrarme
He olvidado mis datos de acceso

Inicio Quiénes somos Tutoriales Formación Comparador de salarios Nuestro libro Charlas Más



DESARROLLADO POR:
Carlos León Villamayor

Consultor tecnológico de desarrollo de proyectos informáticos. Constructor de Adictos Al Trabajo 2.0

Ingeniero Técnico en Informática (cursando grado superior)

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Ver tutoriales de Carlos León Villamayor

Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2012-01-02

5

| Share Regístrate para votar

Primeros pasos con ServiceMix 4.4.0

0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Arquitectura
- 4. Instalación
- 5. Shell y web de Apache Karaf
- 6. Servicio de prueba
- 7. Conclusiones

1. Entorno

Este tutorial está desarrollado usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 2Ghz Intel Core i7 (4 núcleos) 8Gb de RAM
- Sistema Operativo: Mac OS X 10.7.2 (Lion)
- Versión de java SDK 6 instalada en el sistema

2. Introducción

En este tutorial veremos a groso modo este potente ESB (Enterprise Service Bus) opensource y el objetivo será familiarizarnos con el uso y las tecnologías que utiliza.

Las principales caracterÃsticas, es que está basado en componentes opensource totalmente actualizados:

- Sistema de mensajería basado en Apache ActiveMQ
- Mensajería, enrutado y Patrones de Integración con Apache Camel
- WS-* and RESTful web services con Apache CXF
- Integración flexible entre componentes mediante Apache ServiceMix NMR (Normalized Message Router)
- Motor de WS-BPEL con Apache ODE
- Servidor OSGi basado en Apache Karaf

3. Arquitectura

La arquitectura actual es la siguiente:

Últimas Noticias

Todo preparado para el estreno de Terrakas 1x02

¡¡¡Confirmado el jamón para el estreno del segundo capítulo de Terrakas!!!

¡¡¡Terrakas en Antena 3!!!

AdictosAlTrabajo, en la lista final de candidatos a los III Premios Focus al Conocimiento Libre

TERRAKAS - Estrenamos el segundo capítulo!!!

Histórico de NOTICIAS

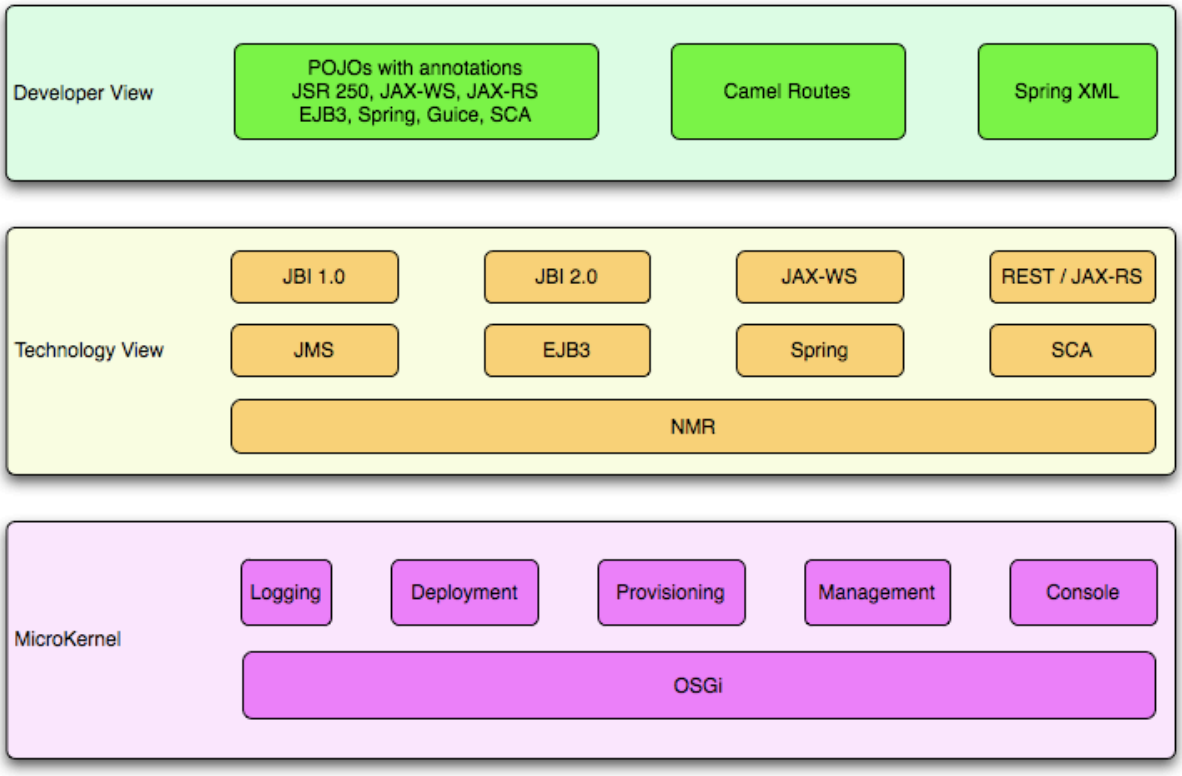
Últimos Tutoriales

Cluster en Glassfishv3 usando mod_jk para el balanceo de peticiones

Introducción a Apache Synapse ESB.

Ejecución de tareas asíncronas y planificadas con Spring.

Monitoriza tu cache con Jconsole activando Jmx con Ehcache integrado con

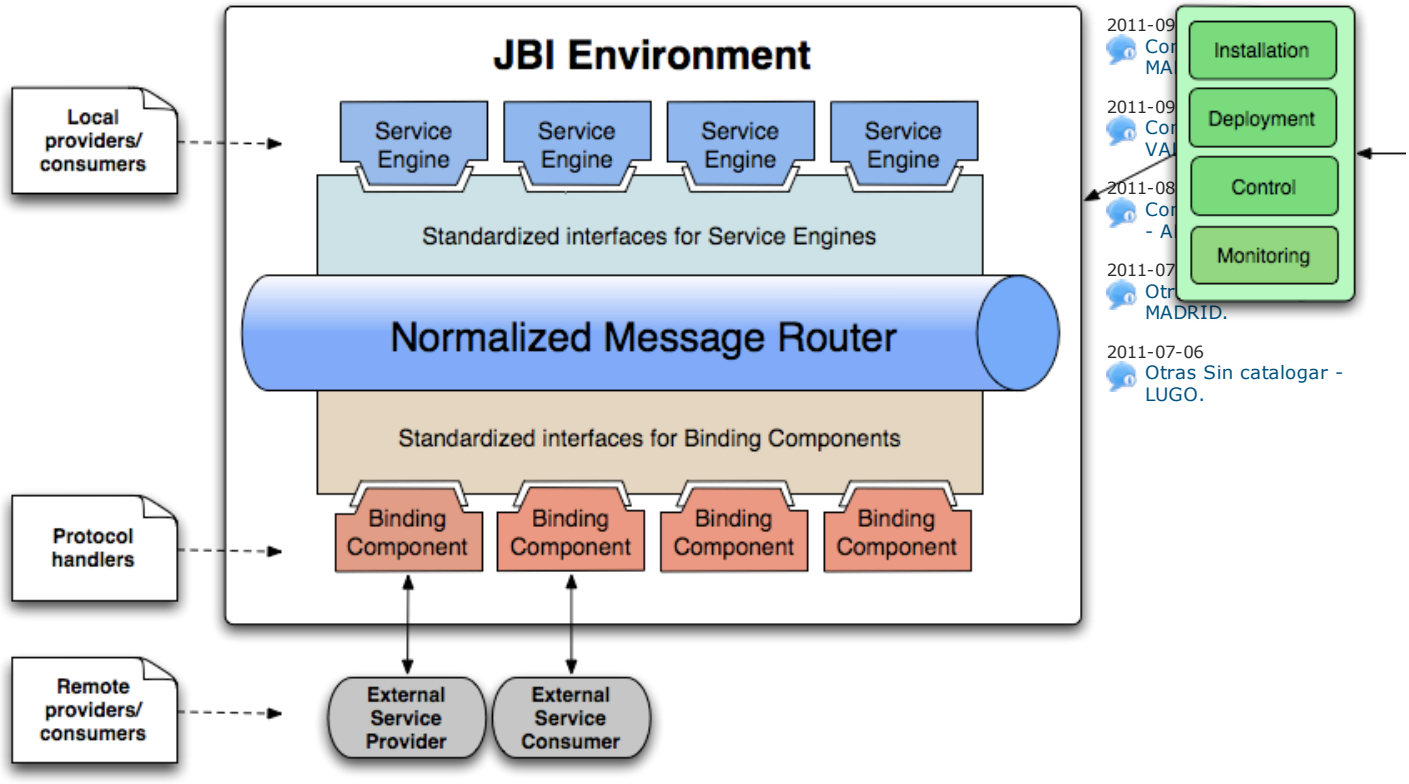


Donde podemos apreciar los diferentes componentes comentados en la introducción.

Principalmente podemos construir servicios basados en patrones de integración de dos formas posibles, utilizando Apache Camel o utilizando los componentes JBI que lleva incorporados. JBI o Java Business Integration pretendía ser el estandar ESB para integración de servicios, pero su primera versión llegó tarde y la mayoría de ESBs de pago así como algunos open source no lo implementan. Actualmente tiene dos versiones (Servicemix es compatible con ambas), aunque la versión 2.0 está muy verde:

- JBI 1.0 (JSR-208)
- JBI 2.0 (JSR-312)

La arquitectura de JBI es la siguiente:



Podemos destacar los siguientes componentes:

- Mensajería basada en WSDL 1.1 o 2.0
- NMR o Normalized Message Router: es el componente principal de JBI, ya que es el encargado de

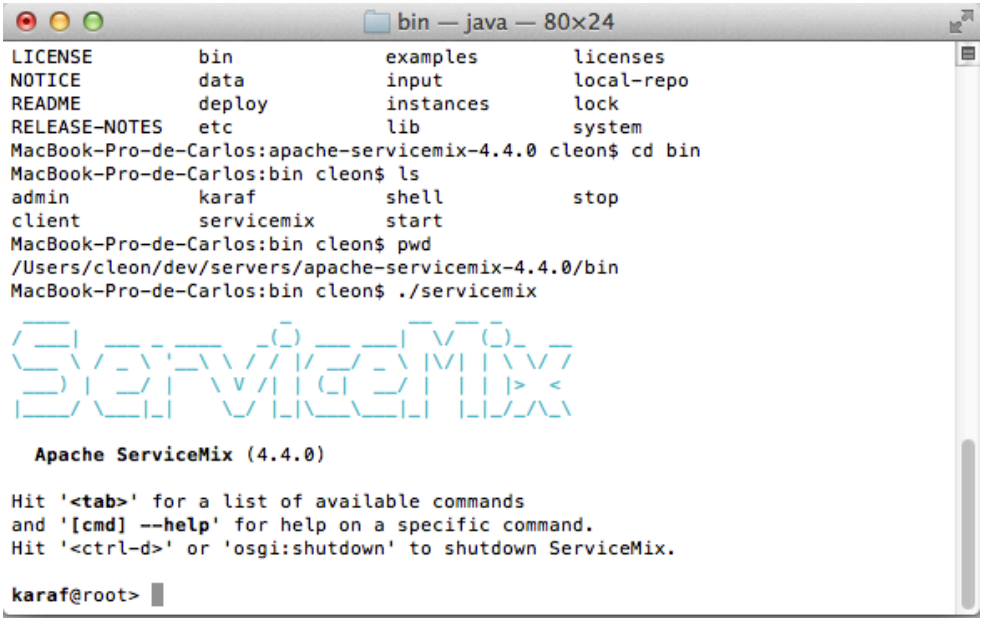
realizar el intercambio de mensajes en todos los componentes, a groso modo, el manejador del bus.
- Service Engines: proveen la lógica de negocio y/o transformaciónes o consumidores de otros servicios.
- Binding Components: proveen la conectividad con servicios externos a ServiceMix
- JMX: permite la instalación de componentes, control y gestiÃ³n del ciclo de vida de servicios entre otros. Con la inclusión de Apache Karaf, la mayoría de estas tareas se pueden realizar por consola, aunque eso lo veremos mas adelante.

4. Instalación

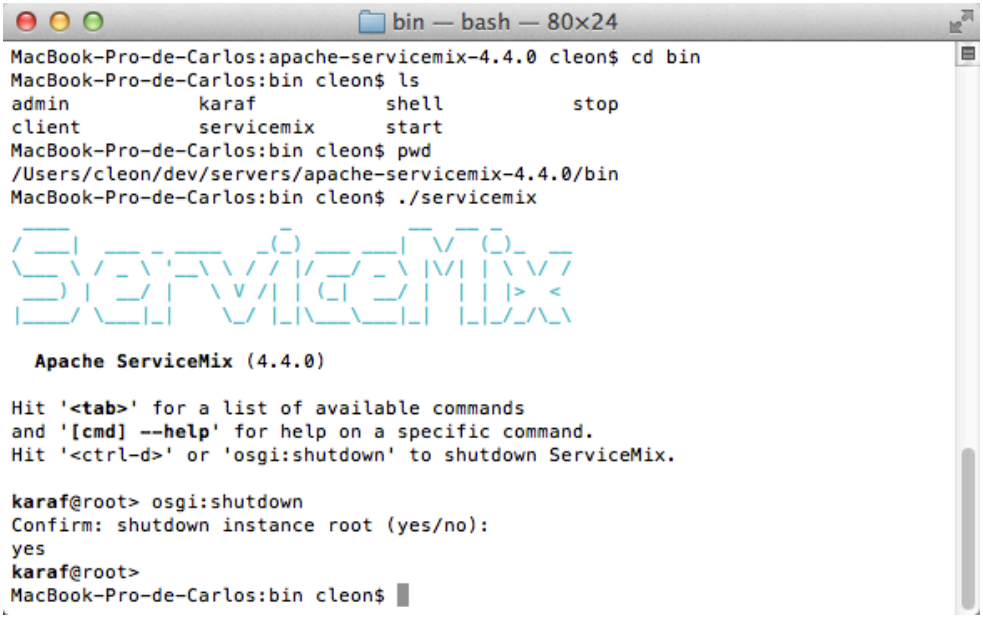
Primero descargaremos la versión 4.4.0 desde la web de ServiceMix: http://servicemix.apache.org/downloads.html. En mi caso será la versión para mac, aunque a nivel de gestión este tutorial es válido para plataformas windows (simplemente varían los ejecutables). Una vez descargado lo descomprimiremos abriendo una consola:

```
MacBook-Pro-de-Carlos:servers cleon$ tar -xvzf apache-servicemix-4.4.0.tar.gz
```

una vez descomprimido, ejecutamos desde la consola el archivo "<SERVICEMIX_HOME>/bin/servicemix" y obtendremos una ventana como la siguiente:



Para apagarlo, tal y como indica en la consola tendremos que escribir **osgi:shutdown** en la consola Karaf. Nos pedirá confirmación y saldremos de la consola y apagaremos servicemix:



Con esto ya hemos aprendido lo más básico de servicemix, en el siguiente punto veremos como utilizar la lÃnea de comandos de Karaf.

5. Shell y web de Apache Karaf

Lo primero que realizaremos será iniciar servicemix como se indicaba en el punto anterior. Lo primero que haremos será introducir el comando "osgi:list" que nos permitirá realizar un listado de los "bundle" disponibles:

```
karaf@root> osgi:list
START LEVEL 100 , List Threshold: 50
```

```
   ID   State         Blueprint      Spring    Level  Name
[  43] [Active     ] [Created     ] [       ] [   60] activemq-broker.xml (0.0.0)
[  45] [Active     ] [            ] [       ] [   60] kahadb (5.5.1)
[  46] [Active     ] [            ] [       ] [   60] activemq-pool (5.5.1)
[  47] [Active     ] [            ] [       ] [   60] activemq-console (5.5.1)
[  48] [Active     ] [            ] [       ] [   60] Apache Aries Transaction Manager (0.2.0.incubating)
[  49] [Active     ] [            ] [       ] [   60] activemq-ra (5.5.1)
[  50] [Active     ] [Created     ] [       ] [   60] activemq-core (5.5.1)
                                      Fragments: 61
[  51] [Active     ] [Created     ] [       ] [   60] activemq-karaf (5.5.1)
[  52] [Active     ] [            ] [       ] [   60] geronimo-j2ee-management_1.1_spec (1.0.1)
[  53] [Active     ] [            ] [       ] [   60] Commons Pool (1.5.4)
[  54] [Active     ] [            ] [       ] [   60] geronimo-jms_1.1_spec (1.1.1)
[  55] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: commons-codec (1.3.0.3)
[  56] [Active     ] [            ] [       ] [   60] Commons Collections (3.2.1)
[  57] [Active     ] [            ] [       ] [   60] Apache ServiceMix Bundles: velocity-1.6.2 (1.6.2.3)
[  58] [Active     ] [            ] [       ] [   60] Apache ServiceMix Bundles: oro-2.0.8 (2.0.8.3)
[  59] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: jasypt (1.6.0.1)
[  60] [Active     ] [            ] [       ] [   60] Apache ServiceMix Bundles: commons-lang-2.4 (2.4.0.3)
[  61] [Resolved   ] [            ] [       ] [   60] activemq-blueprint (5.5.1)
                                      Hosts: 50
[  62] [Active     ] [Created     ] [       ] [   60] Apache XBean :: OSGI Blueprint Namespace Handler (3.7)
[  63] [Active     ] [            ] [       ] [   60] Commons JEXL (2.0.1)
[  64] [Active     ] [            ] [       ] [   60] Apache ServiceMix Specs :: Scripting API 1.0 (1.5.0)
[  77] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: cglib (2.1.0.3_7)
[  78] [Active     ] [            ] [       ] [   60] activemq-spring (5.5.1)
[  79] [Active     ] [            ] [       ] [   60] Apache XBean :: Spring (3.7)
[  90] [Active     ] [            ] [       ] [   50] camel-core (2.8.3)
[  91] [Active     ] [            ] [       ] [   50] Commons Management (1.0)
[  92] [Active     ] [Created     ] [       ] [   50] camel-karaf-commands (2.8.3)
[  94] [Active     ] [            ] [       ] [   50] camel-spring (2.8.3)
[  95] [Active     ] [            ] [       ] [   60] Jetty :: Asynchronous HTTP Client (7.4.5.v20110725)
[  96] [Active     ] [            ] [       ] [   60] Jetty :: Server Core (7.4.5.v20110725)
[  97] [Active     ] [            ] [       ] [   60] Jetty :: Servlet Handling (7.4.5.v20110725)
[  98] [Active     ] [            ] [       ] [   60] Jetty :: IO Utility (7.4.5.v20110725)
[  99] [Active     ] [            ] [       ] [   60] Jetty :: Security (7.4.5.v20110725)
[ 100] [Active     ] [            ] [       ] [   60] Jetty :: Http Utility (7.4.5.v20110725)
[ 101] [Active     ] [            ] [       ] [   60] Jetty :: Continuation (7.4.5.v20110725)
[ 102] [Active     ] [            ] [       ] [   60] Jetty :: JNDI Naming (7.4.5.v20110725)
[ 103] [Active     ] [            ] [       ] [   60] Jetty :: Webapp Application Support (7.4.5.v20110725)
[ 104] [Active     ] [            ] [       ] [   60] Jetty :: Utility Servlets and Filters (7.4.5.v20110725)
[ 105] [Active     ] [            ] [       ] [   60] Jetty :: Plus (7.4.5.v20110725)
[ 106] [Active     ] [            ] [       ] [   60] Jetty :: JMX Management (7.4.5.v20110725)
[ 107] [Active     ] [            ] [       ] [   60] Jetty :: XML utilities (7.4.5.v20110725)
[ 108] [Active     ] [            ] [       ] [   60] Jetty :: Utilities (7.4.5.v20110725)
[ 109] [Active     ] [            ] [       ] [   60] geronimo-servlet_2.5_spec (1.1.2)
[ 110] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - Service SPI (1.0.7)
[ 111] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - Jetty (1.0.7)
[ 112] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - API (1.0.7)
[ 113] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - Runtime (1.0.7)
[ 118] [Active     ] [            ] [       ] [   50] camel-jms (2.8.3)
[ 119] [Active     ] [            ] [       ] [   50] activemq-camel (5.5.1)
[ 122] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - Extender - WAR (1.0.7)
[ 123] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - FileInstall Deployer (1.0.7)
[ 124] [Active     ] [            ] [       ] [   60] OPS4J Pax Url - war (1.2.8)
[ 125] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - Extender - Whiteboard (1.0.7)
[ 126] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: opensaml (2.5.1.1)
[ 127] [Active     ] [            ] [       ] [   60] Joda-Time (1.6.2)
[ 128] [Active     ] [            ] [       ] [   60] WSS4J (1.6.3)
[ 129] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: xmlsec (1.4.5.1)
[ 130] [Active     ] [            ] [       ] [   60] Axiom API (1.2.10)
[ 131] [Active     ] [            ] [       ] [   60] Abdera Parser (1.1.2)
[ 132] [Active     ] [            ] [       ] [   60] Axiom Impl (1.2.10)
[ 133] [Active     ] [            ] [       ] [   60] Abdera Extensions - Main (1.1.2)
[ 134] [Active     ] [            ] [       ] [   60] I18N Libraries (1.1.2)
[ 135] [Active     ] [            ] [       ] [   60] Abdera Core (1.1.2)
[ 137] [Active     ] [Created     ] [       ] [   60] Apache CXF Bundle Jar (2.4.4)
[ 138] [Active     ] [            ] [       ] [   60] jettison (1.3)
[ 139] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: js (1.0.0.7R2_1)
[ 140] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: xmlbeans (2.4.0.4)
[ 142] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: xmlresolver (1.2.0.3)
[ 144] [Active     ] [            ] [       ] [   60] Commons Lang (2.6)
[ 146] [Active     ] [            ] [       ] [   60] Apache Neethi (3.0.1)
[ 147] [Active     ] [            ] [       ] [   60] XmlSchema Core (2.0.1)
[ 148] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: wsdl4j (1.6.2.3)
[ 149] [Active     ] [            ] [       ] [   60] Apache ServiceMix :: Bundles :: jdom (1.1.0.3)
[ 150] [Active     ] [            ] [       ] [   50] camel-cxf-transport (2.8.3)
[ 151] [Active     ] [Created     ] [       ] [   50] camel-cxf (2.8.3)
[ 152] [Active     ] [Created     ] [       ] [   50] camel-blueprint (2.8.3)
[ 153] [Active     ] [            ] [       ] [   60] OPS4J Pax Web - Jsp Support (1.0.7)
[ 156] [Active     ] [            ] [       ] [   60] camelinputoutput.xml (0.0.0)
[ 157] [Active     ] [            ] [       ] [   60] Apache Felix Metatype Service (1.0.4)
[ 158] [Resolved   ] [            ] [       ] [   60] Apache Karaf :: Web Console :: Branding (2.2.4)
                                      Hosts: 159
[ 159] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Console (2.2.4)
                                      Fragments: 158
```

```
[ 160] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Admin Plugin (2.2.4)
[ 161] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Features Plugin (2.2.4)
[ 162] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Gogo Plugin (2.2.4)
[ 163] [Active     ] [            ] [       ] [   60] Apache Felix Web Console Event Plugin (1.0.2)
karaf@root>
```

Donde podemos observar una tabla con los siguientes componentes:

- ID: el identificador del bundle
- State: el estado del bundle
- Blueprint/Spring: mostrarán si los bean se han creado correctamente.
- Level: muestra el orden de arranque
- Name: muestra el nombre del bundle

Apache Karaf trabaja con bundles y posee las siguientes características:

- Despliegue en caliente: de OSGi bundles monitorizando archivos jar en el directorio "<SERVICEMIX_HOME>/deploy". También soporta "exploded bundles" (jar descomprimidos) y descriptores propios (blueprint y spring).
- Configuración Dinámica: todos los servicios se configuran habitualmente a través del servicio OSGi ConfigurationAdmin. Dicha configuración se puede modificar utilizando archivos de propiedades dentro del directorio "<SERVICEMIX_HOME>/etc directory". Esta configuración está monitorizada y los cambios realizados se propagaran a los servicios en caliente.
- Centralización del Log: utilizando un backend de log centralizado por log4j, soporta variedad de APIs (JDK 1.4, JCL, SLF4J, Avalon, OSGi)
- Provisionamiento: la provisiÃ³n de bibliotecas o aplicaciones puede realizarse de diferentes formas.
- Integración nativa con el SSOO huesped: puede integrarse como servicio dentro del SSOO.
- Consola de comandos extensible: dispone de una interfaz de comandos desde la que se pueden realizar casi todas las tareas de mantenimiento de servicios, así como de la plataforma.
- Acceso remoto: se puede utilizar ssh para enviar comandos a la consola.
- Seguridad basada en JAAS

Tal y como se mostraban en las últimas lÃneas del los bundles instalados en el sistema:

```
...
[ 158] [Resolved   ] [            ] [       ] [   60] Apache Karaf :: Web Console :: Branding (2.2.4)
                                        Hosts: 159
[ 159] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Console (2.2.4)
                                        Fragments: 158
[ 160] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Admin Plugin (2.2.4)
[ 161] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Features Plugin (2.2.4)
[ 162] [Active     ] [Created     ] [       ] [   60] Apache Karaf :: Web Console :: Gogo Plugin (2.2.4)
[ 163] [Active     ] [            ] [       ] [   60] Apache Felix Web Console Event Plugin (1.0.2)
karaf@root>
```

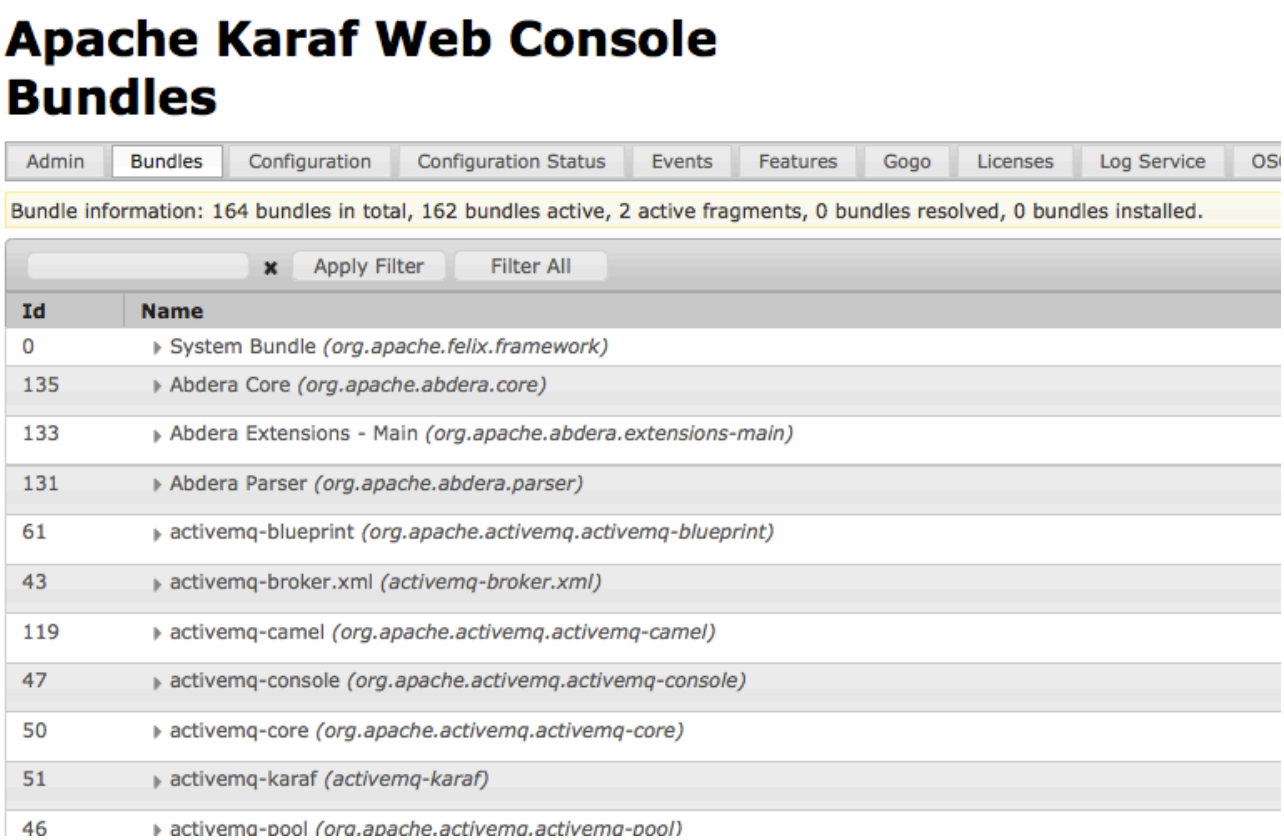
Tengo instalada y activa una consola web, que se puede instalar mediante el comando:

```
karaf@root> features:install webconsole
```

Una vez instalada podemos ver su estado mediante el comando:

```
karaf@root> features:list | grep webconsole
[installed  ] [2.2.4          ] webconsole-base                    karaf-2.2.4
[installed  ] [2.2.4          ] webconsole                         karaf-2.2.4
```

Para ver la consola web abrimos un navegador con la siguiente ruta http://localhost:8181/system/console
donde por defecto tendrá el usuario/password smx. Deberemos ver una pantalla parecida a esta:



Desde esta consola web podremos realizar prácticamente lo mismo que desde la consola de comandos.

Otros comandos interesantes son:

- log:display: muestra las últimas líneas de log
- log:display-exception: muestra la última excepcion producida
- log:set NIVEL: establece el nivel de log que se mostrará
- log:tail: muestra el log en tiempo de ejecución modificandose cuando se produzcan cambios en la salida de log

6. Servicio de prueba

Una vez vista por encima la arquitectura y las caracterÃsticas de ServiceMix, vamos a hacer un ejemplo muy simple, un router que enviará un archivo de un directorio a otro (en mi caso, desde "/Users/cleon/input" a "/Users/cleon/output"). Para ello escribiremos el siguiente archivo xml:

camelinputoutput.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:camel="http://camel.apache.org/schema/spring"
       xsi:schemaLocation="
            http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
                        http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring-2.8.3.xs
        <camelContext xmlns="http://camel.apache.org/schema/spring">
                <route>
                        <from uri="file:/Users/cleon/input"/>
                        <log message="Copiando ${file:name} al directorio de salida"/>
                        <to uri="file:/Users/cleon/output"/>
                </route>
        </camelContext>
</beans>
```

Y lo copiamos al directorio "<SERVICEMIX_HOME>/deploy". Si observamos el log veremos que sÃ se ha desplegado correctamente:

```
karaf@root> log:clear
karaf@root> log:display
2011-12-30 14:51:59,081 | INFO  | mix-4.4.0/deploy | ultOsgiApplicationContextCreator | ?
2011-12-30 14:51:59,081 | INFO  | ExtenderThread-7 | OsgiBundleXmlApplicationContext  | ?
2011-12-30 14:51:59,091 | INFO  | ExtenderThread-7 | XmlBeanDefinitionReader          | ?
2011-12-30 14:51:59,163 | INFO  | ExtenderThread-7 | WaiterApplicationContextExecutor | ?
2011-12-30 14:51:59,165 | INFO  | ExtenderThread-8 | DefaultListableBeanFactory       | ?
2011-12-30 14:51:59,171 | INFO  | ExtenderThread-8 | OsgiSpringCamelContext           | ?
2011-12-30 14:51:59,172 | INFO  | ExtenderThread-8 | OsgiSpringCamelContext           | ?
2011-12-30 14:51:59,207 | INFO  | ExtenderThread-8 | AnnotationTypeConverterLoader    | ?
2011-12-30 14:51:59,215 | INFO  | ExtenderThread-8 | DefaultTypeConverter             | ?
2011-12-30 14:51:59,217 | INFO  | ExtenderThread-8 | Activator                        | ?
2011-12-30 14:51:59,218 | INFO  | ExtenderThread-8 | Activator                        | ?
```

```
2011-12-30 14:51:59,545 | INFO  | ExtenderThread-8 | OsgiSpringCamelContext           | ?
2011-12-30 14:51:59,548 | INFO  | ExtenderThread-8 | OsgiSpringCamelContext           | ?
2011-12-30 14:51:59,549 | INFO  | ExtenderThread-8 | OsgiSpringCamelContext           | ?
2011-12-30 14:51:59,550 | INFO  | ExtenderThread-8 | OsgiBundleXmlApplicationContext  | ?
2011-12-30 14:51:59,550 | INFO  | ExtenderThread-8 | ContextLoaderListener            | ?

karaf@root>
```

Ahora en mi caso copiaré el mismo xml de descripción del servicio al directorio de entrada. Si todo ha ido bien, veremos que rapidamente mueve el archivo al directorio destino. Visualizando los logs en la consola:

```
karaf@root> log:display
2011-12-30 14:57:10,281 | INFO  | sers/cleon/input | route3                          | ?

karaf@root>
```

Con esto ya tenemos un ejemplo básico sobre servicemix.

7. Conclusiones

Este es el primer tutorial de ServiceMix que iremos complementando con el tiempo, ya que se trata de un producto muy potente con muchos componentes y dada su escasa documentación (en las versiones 3.X ya era escasa, ahora en la 4.X lo es más) el proceso de aprendizaje es mas complejo. Aun así, animo a utilizarlo, ya que es un componente muy potente.

Cualquier duda o sugerencia podÃ©is comentarlo.

Saludos.

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

| Share

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

SOME RIGHTS RESERVED Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

Copyright 2003-2012 © All Rights Reserved | Texto legal y condiciones de uso | Banners | Powered by Autentia | Contacto

W3C XHTML 1.0 W3C CSS XML RSS XML ATOM