



Servicios de Mensajes en JEE

Sesión3: JMS en aplicaciones reales (2)



Indice

- Esquema de la aplicación
- Desarrollo:
 - Envío de mensajes XML
 - MDB: Parsing de mensajes XML
 - EJB: recibe resultado y envía mensaje XML
 - Cliente consulta el resultado en otra cola



Indice

- Esquema de la aplicación
- Desarrollo:
 - Envío de mensajes XML
 - MDB: Parsing de mensajes XML
 - EJB: recibe resultado y envía mensaje XML
 - Cliente consulta el resultado en otra cola



Esquema de la aplicación

1. Un cliente construye y envía una petición a una cola.
2. La petición es recogida (modo asíncrono) por un bean de mensajes (MDB).
3. El MDB invoca (síncrono) a un EJB de sesión para realizar la petición del cliente.
4. El EJB devuelve el resultado al bean de mensajes.
5. El bean empaqueta el resultado en un mensaje y lo envía a otra cola.
6. Un segundo cliente recoge la respuesta.

Integración de componentes + procesamiento XML



Indice

- Esquema de la aplicación
- Desarrollo:
 - Envío de mensajes XML
 - MDB: Parsing de mensajes XML
 - EJB: recibe resultado y envía mensaje XML
 - Cliente consulta el resultado en otra cola



Envío de Mensajes XML y MDB

- `ClientSend.java` (en WL 9.2)
 - **Funcionamiento:**
 - Construye un mensaje como el siguiente:
`<stocktrade action=buy symbol=BEAS numShares=10 \>`
 - Para ello usa un String con los datos que va pidiendo al usuario. Después envía el string como mensaje XML.
- `MessageTraderBean.java`
 - **Funcionamiento:**
 - MDB que recoge asíncronamente el mensaje.
 - Lo interesante es que en su `onMessage()` lo parsea creando un `RequestHandler` y un `SAXParser`.



MDB y EJB

- `MessageTraderBean.java`

- **Funcionamiento:**

- Dentro de `onMessage()` invoca a un EJB para que realice la acción parseada. Ya tiene los datos necesarios para invocarlos métodos del EJB.

- `TraderBean.java`

- **Funcionamiento:** Se trata de un EJB de sesión

```
@EnvEntries({  
    @EnvEntry(name = "BEAS", type = "java.lang.Double",  
        value="100.0")  
    @EnvEntry(name = "MSFT", type = "java.lang.Double",  
        value = "150.0")  
})
```



MDB y Cliente

- `MessageTraderBean.java`
 - **Funcionamiento:**
 - Dentro de `onMessage()` recibe el resultado de la operación y envía un mensaje XML a una cola
- `ClientReceive.java`
 - **Funcionamiento:**
 - El cliente java recoge el mensaje XML y lo parsea de nuevo.



Ejercicios...

- **Aplicación completa.**

Compilad, desplegad y probad el ejemplo

```
examples.jms.messageformat;
```

- **Ejemplo de otra aplicación.**

Compilad, desplegad y probad el ejemplo de aplicación motivadora que se planteó al principio del módulo.

De forma optativa, refactorizarla para uso de EJB.